Proceedings of the

Multigrid Tutorial, with Applications to Molecular Dynamics

October 10–12, 1995. Weizmann Institute, Rehovot, Israel

*** Workshop sponsor**

The workshop was sponsored by the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science, Rehovot, Israel.

*** Acknowledgement**

We wish to thank the United States Air Force European Office of Aerospace Research and Development for its contribution to the success of this workshop.

Workshop Organizers:

Achi Brandt,
Weizmann Institute of Science
Tamar Schlick,
Courant Institute of Mathematics

Secretary:

Carol Weintraub
Office: 972-8-343545
Fax: 972-8-344122
Email: carol@wisdom.weizmann.ac.il

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1995	3. REPORT TYPE AND DATES COVERED Conference Proceedings

4. TITLE AND SUBTITLE Workshop on Multigrid Tutorial, with Applications to Molecular Dynamicss	5. FUNDING NUMBERS F6170895W0467
6. AUTHOR(S) Conference Committee	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science Rehovot Rehovot 76100 Israel	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD PSC 802 BOX 14 FPO 09499-0200	10. SPONSORING/MONITORING AGENCY REPORT NUMBER CSP 95-1048

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE A

13. ABSTRACT (Maximum 200 words)

The Final Proceedings for Multigrid Tutorial, with Applications to Molecular Dynamics, 10 October 1995 - 12 October 1995

14. SUBJECT TERMS Nil	15. NUMBER OF PAGES 233 16. PRICE CODE N/A

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19, SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

Table of Contents

Page

Lecture Transparencies:

WORKSHOP PROGRAM -- Multigrid and Molecular Dynamics

October 10-12, 1995

Dept. of Applied Mathematics & Computer Science
The Weizmann Institute of Science, Ziskind Building,
Lecture Room 1 (basement level)

Note: The program may change even during the course itself according to the interest of participants.

Tuesday, October 10th

8:30 - 9:15	Registration, organization, coffee
9:00 - 9:15	Introductory remarks
9:15 - 10:30	Introduction: tasks and types of multiscale computation (Brandt)
10:30 - 11:00	Coffee
11:00 - 12:15	Multiscale decomposition of forces and fast force summation (Brandt)
12:15 - 13:30	Lunch
13:30 - 15:00	Elements of linear multigrid for PDEs (Stueben)
15:00 - 15:30	Coffee
15:30 - 17:00	Poster session, including: -Gomathi Ramachandran: Buckling transitions in superhelical DNA: dependence on the elastic constants and DNA size -Bimalendu Mishra: Error analysis in Langevin dynamics simulations -Margaret Mandziuk: Resonance in the dynamics of chemical systems -Eric Barth: LIN and a family of related methods for molecular dynamics simulation -Hongmei Jian: Computer simulation of DNA using a discrete worm-like chain model -Gerd Winter: BEMOLPA project
19:30	Mini-vans depart for dinner at Pearl of the Sea restaurant, in Rishon L'Tzion

Wednesday, October 11th

8:45 - 9:00	Coffee
9:00 - 10:30	Multigrid Monte-Carlo and stochastic coarsening (Brandt; Mack?)
10:30 - 11:00	Coffee
11:00 - 12:15	Global and discrete-state optimization: multiscale annealing (Brandt)
12:15 - 14:00	Lunch
14:00 - 15:00	Full multigrid and nonlinear multigrid (Stueben)
15:00 - 15:30	Eric Barth and Margaret Mandziuk: Molecular dynamics simulation at large timesteps
15:30 - 16:00	Coffee
16:00 - 16:30	Bob Skeel: Efficient use of fast electrostatics in molecular dynamics
16:30 - 17:00	Chris Lambert: Efficient dense Hessian computation in molecular minimization
17:00 - 17:30	Oren Becker: Reduced variable molecular dynamics
17:30 - 18:00	Dexuan Xie: A remark on algebraic multigrid analysis
19:00	Dinner in town at one's choice

Thursday, October 12th

8:45 - 9:00	Coffee
9:00 - 10:00	Algebraic multigrid (Stueben, Ruge)
10:00 - 10:30	Coffee
10:30 - 12:00	Multiscale methods in molecular dynamics (Bai, Brandt)
12:00 - 14:00	Discussion on future directions; box lunch
16:00	Bus departs for dinner in Yaffo, and evening at Tel Aviv seaside promenade

List of Participants

Dov Bai
Cornell University
Ithaca, NY 14853-7501
USA
bai@msiadmin.cit.cornell.edu

Eric Barth
Courant Institute of Mathematics
New York University
251 Mercer St.
New York, NY 10021
USA
barthe@cims.nyu.edu
(and Howard Hughes Medical Institute)

Oren Becker
School of Chemistry
Tel Aviv University
Ramat Aviv, Tel Aviv 69578
Israel
becker@sapphire.tau.ac.il

Achi Brandt
Dept. of Appl. Math. & Comp. Sci.
Weizmann Institute of Science
Rehovot 76100
Israel
mabrandt@weizmann.weizmann.ac.il

Nathan Dinar
Dept. of Applied Mathematics
Israel Institute of Biological Research
Israel
natan@math.wxiibr.co.il

Boris Diskin
Dept. of Appl. Math. & Comp. Sci.
Weizmann Institute of Science
Rehovot 76100
Israel
diskin@wisdom.weizmann.ac.il

Patrick Fischer
Dept. of Computer Science
School of Mathematical Sciences
Tel Aviv University
Ramat Aviv 69978
Israel
fischer@math.tau.ac.il

Meirav Galun
Dept. of Appl. Math. & Comp. Sci.
The Weizmann Institute of Science
Rehovot 76100
Israel
meirav@wisdom.weizmann.ac.il

Hongmei Jian
Courant Institute of Mathematics
New York University
251 Mercer St.
New York, NY 10021
USA
jianh@acf2.nyu.edu

Jacob Katzenelson
Dept. of Elec. Engr.
Technion
Haifa 32000
Israel
jacob@ee.technion.ac.il

Arnold Krechel
Schloss Birlinghoven
D-53757 St. Augustin
Germany
krechel@gmd.de

Christophe Lambert
Dept. of Computer Science
Duke University
Room D327, LSRC
Durham, NC 27708
USA
lambert@cs.duke.edu

Oren Livne
Dept. of Appl. Math. & Comp. Sci.
The Weizmann Institute of Science
Rehovot 76100
Israel
livneo@wisdom.weizmann.ac.il

Ton Lubrecht
INSA de Lyon
LMC Bâtiment 113-20
20 Avenue Albert Einsein
69621 Villeurbanne Cedex
Lyon, France
Ton.Lubrecht@lmc.insa-lyon.fr

Margaret Mandzuik
Howard Hughes Medical Institute
New York University
New York, NY 10021
USA
mandzk@watson.biomath.nyu.edu

Tamar Naeh
Dept. of Mathematics
Tel Aviv University
Ramat Aviv
Tel Aviv
Israel
naeh@math.tau.ac.il

Ruth Pachter (& Meir Pachter)
Materials Directorate
WL/MLPJ
Wright Patterson AFB
Ohio 45433–6533
USA
pachterr@ml.wpafb.af.mil

Gomathi Ramachandran
Howard Hughes Medical Institute
New York University
New York, NY 10021
USA
ramachan@franklin.biomath.nyu.edu

John Ruge
Dept. of Mathematics
Univ. of Colorado
Boulder, CO
USA
jruge@newton.colorado.edu

David Saada
Dept. of Mathematics
Technion
Haifa 32000
Israel
phr76ds@phjoan2.technion.ac.il

Tamar Schlick
Courant Institute of Mathematics
New York University
251 Mercer Street
New York, NY 10021
USA
schlick@acf4.nyu.edu

Schwichtenberg, Horst
Schloss Birlinghoven
D-53757 St. Augustin
Germany
schwichtenberg@gmd.de

Robert D. Skeel
Dept. of Computer Science, UIUC
1304 West Springfield Avenue
Urbana, Illinois 61801–2987
USA
skeel@cs.uiuc.edu

Klaus Stueben
Gesellschaft fur Mathematik und Datenver
Inst. fur Methodische Grundlagen
Postfach 1316
W-5205 Sankt Augustin 1
Germany
gmap02@f1klaus.gmd.de

Alex Volynski
c/o Dept. of Geology & Mineralogy
Ben Gurion University
P.O.B. 653
Beer Sheva 84105
Israel

Holger Wallmeier
Hoechst AG
Zentralforschung
Scientific Computing
G 865 A
65926 Frankfurt am Main
Germany
wallmeier@msmwia.hoechst.hoechst-ag.d400.de

Winter, Gerd
Schloss Birlinghoven
D-53757 St. Augustin
Germany
Gerd.Winter@gmd.de

Dexuan Xie
Science & Research I, 221c
High Performance Computing Centers
University of Houston
Houston, TX 77204–5502
USA
xie@sina.tcamc.uh.edu

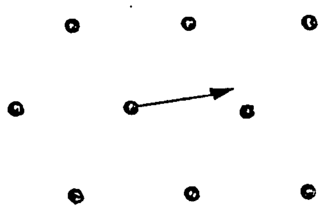
Wei Xu
Courant Institute of Mathematics
New York University
251 Mercer St.
New York, NY 10021
USA
weixu@franklin.biomath.nyu.edu

Leonid Zaslavsky
Dept. of Appl. Math. & Comp. Sci.
The Weizmann Institute of Science
Rehovot 76100
Israel
zasl@wisdom.weizmann.ac.il

A. Brandt

Introduction: Task and Types of Multiscale Computation

- n particles : $n \sim 10^3, 10^{10}, \ldots$
- Calculating $O(n^2)$ forces at each time step
- Extremely short steps



- Wrong attraction basins
- Thermal statistical fluctuations : Need represent many probable u

All addressed by

multi-scale methods

multigrid multi-level multi-resolution
FFT, Renormalization Group,
Wavelets, Multipole, Fractals

Hardware:

Ever smaller and faster

Past: 10 times faster
each ~ 6 years

Future: Parallel processing

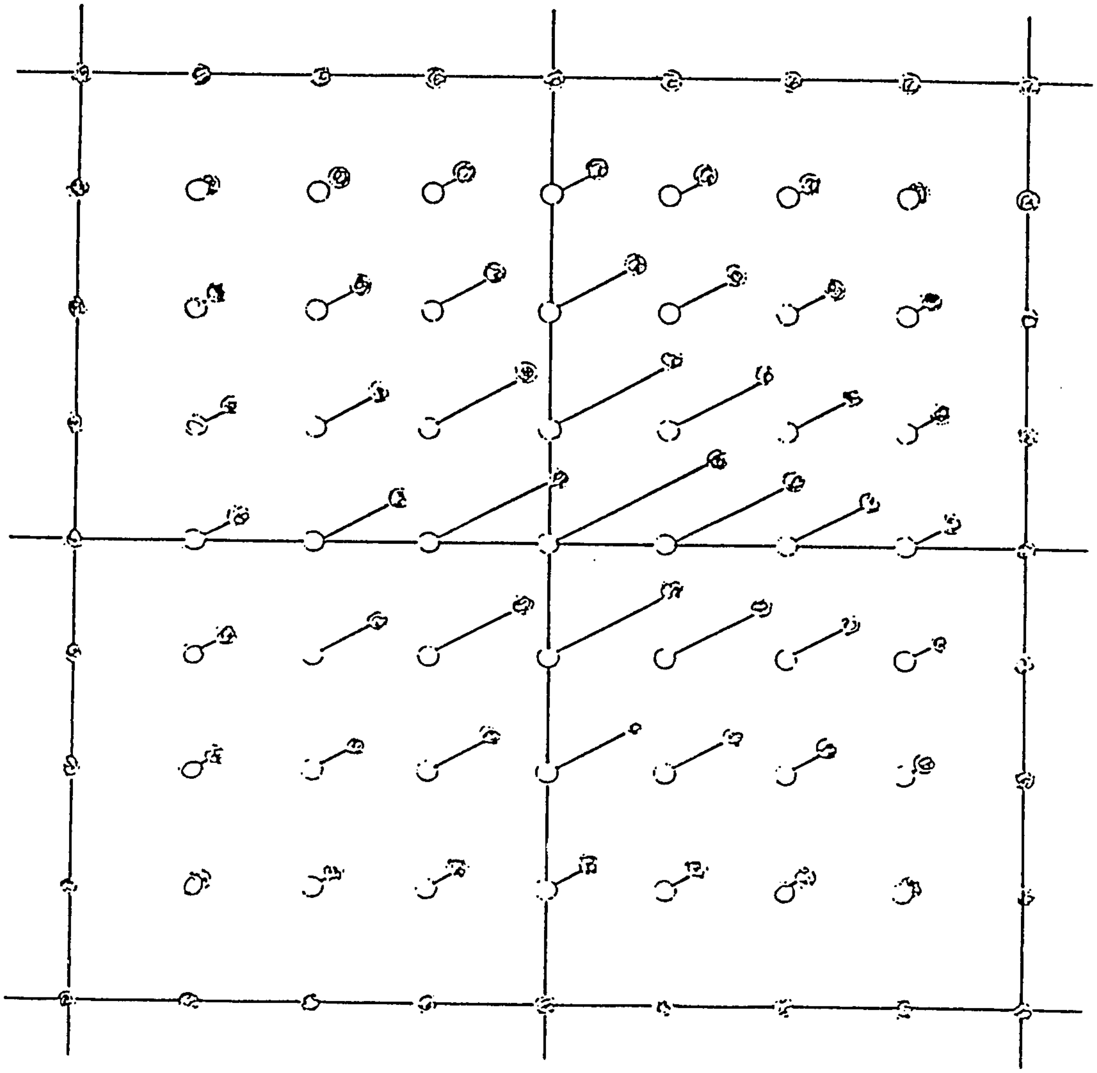
Algorithms

Required amount of computations
rise slower with
increased problem size

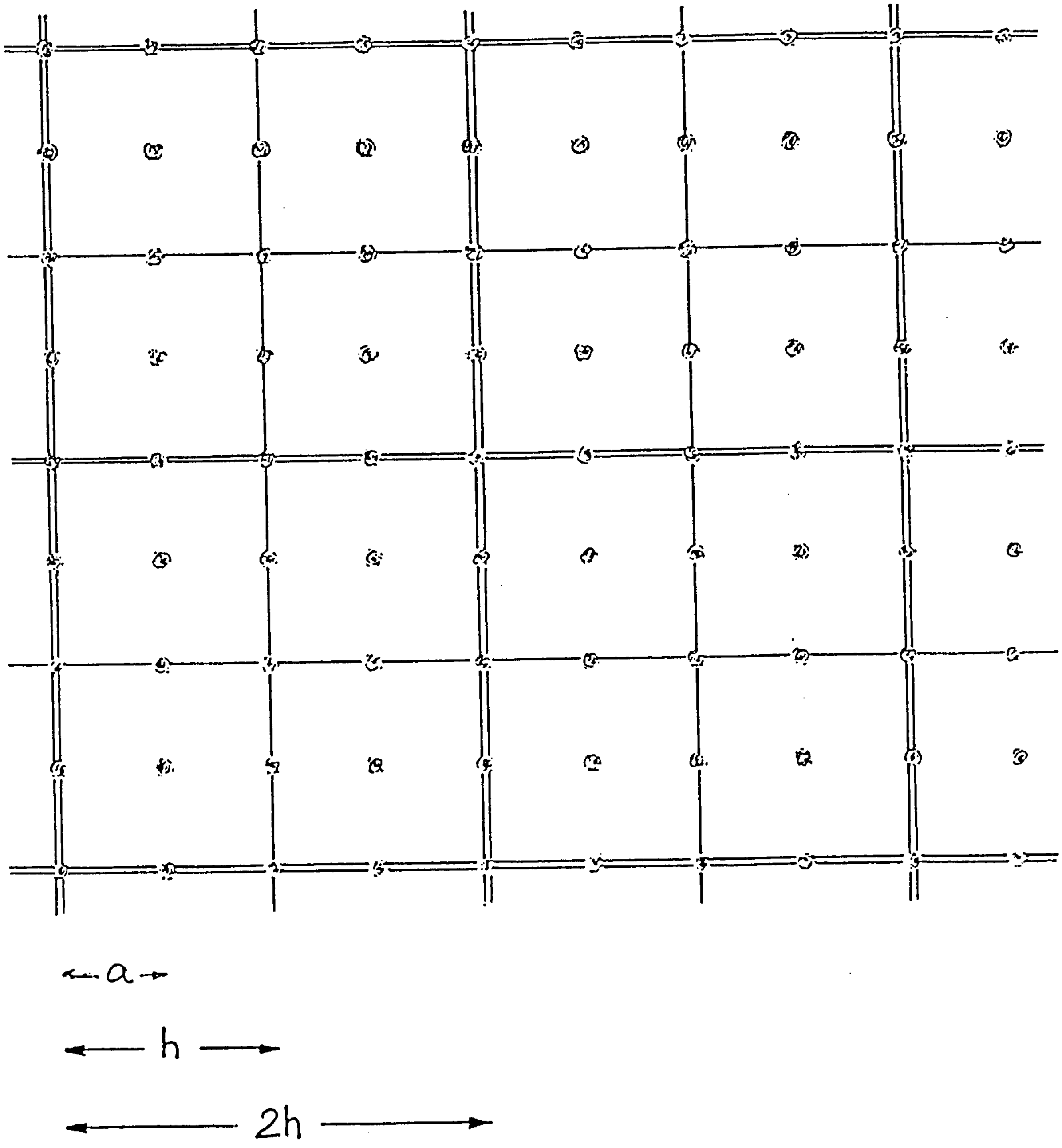
Past: comparable contribution

Future: Multiscale methods

Highly compatible
Excel at large problems



- Similarly : for non-uniform atom distribution.
- Collective force : via the finer-grid forces
  Energy (grid)


a
h
2h

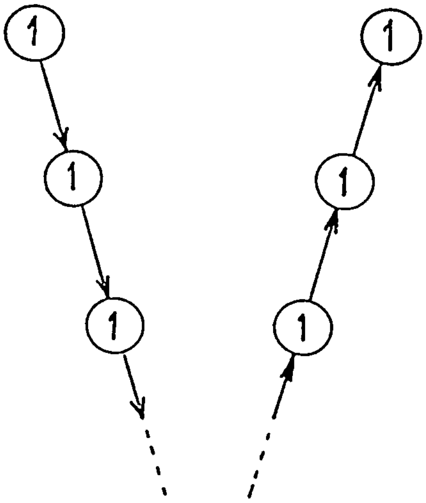
Multigrid cycle

sweeps on

scale a (particles)

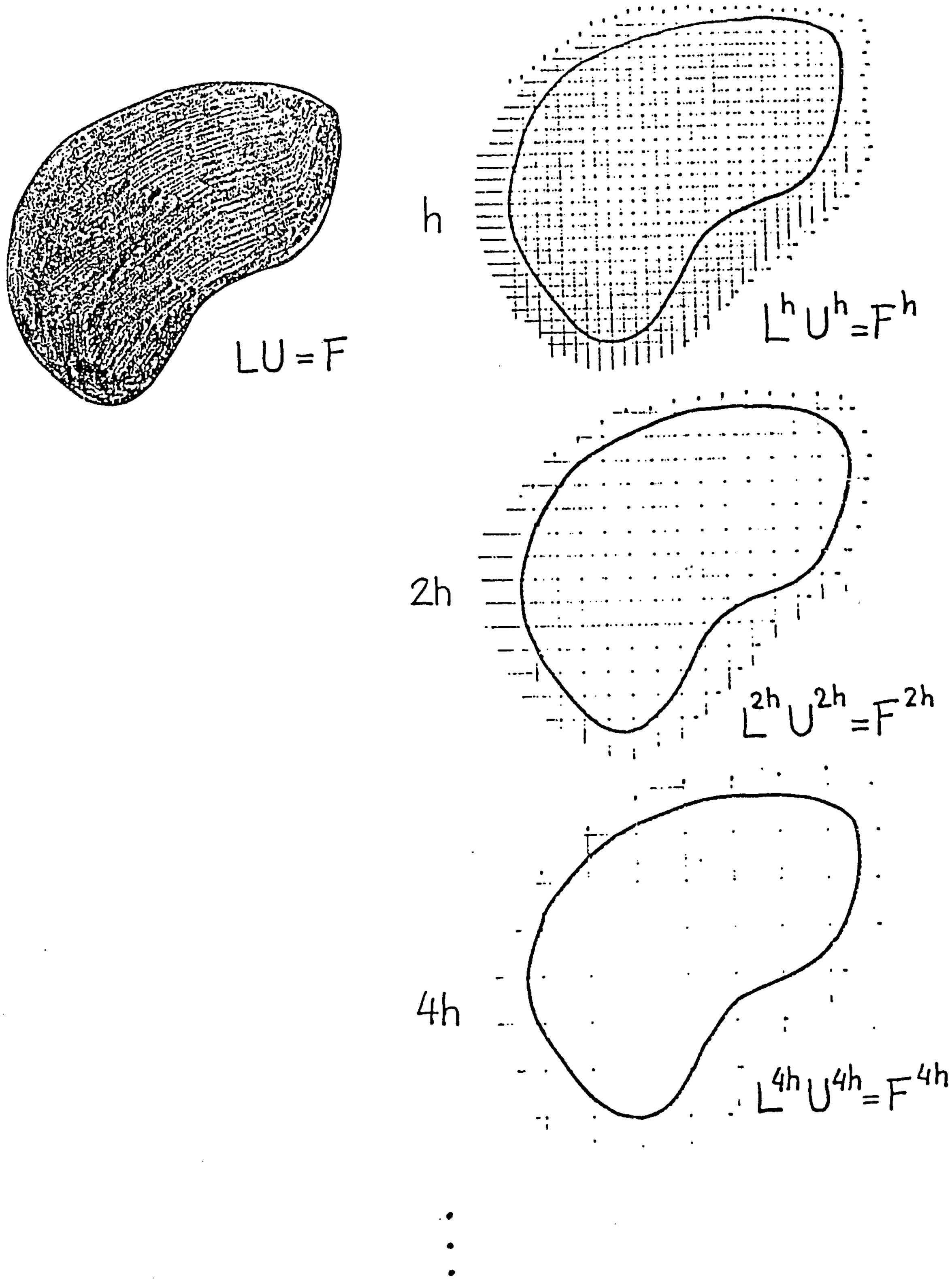
grid h

grid 2h

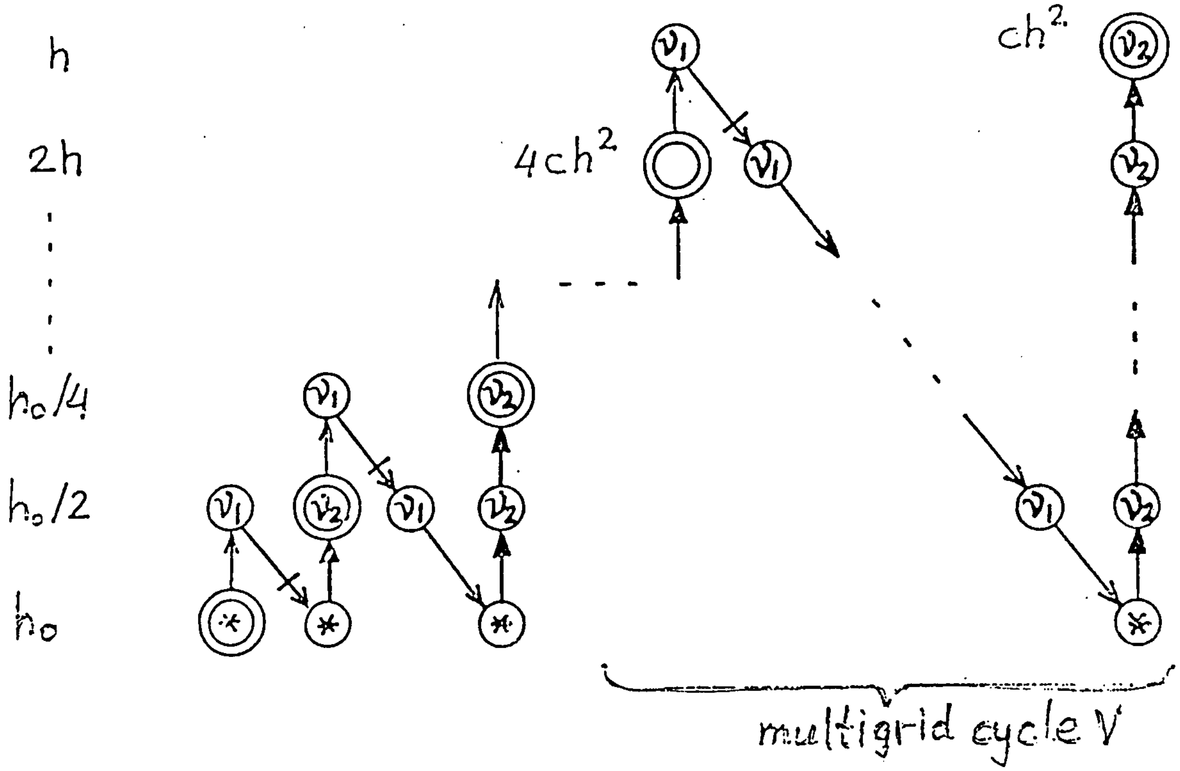


Very large scale changes in one cycle

Negligible work on coarse levels


$LU=F$
h
$L^hU^h=F^h$
2h
$L^{2h}U^{2h}=F^{2h}$
4h
$L^{4h}U^{4h}=F^{4h}$

FULL MULTIGRID (FMG) ALGORITHM



↑ interpolation (order l+p) to a new grid.

▲ interpolation (order m) of corrections:

↘ residual transfer

ⓥ ν relaxation sweeps

◎ algebraic error < truncation error

⊛ enouph sweeps or direct solver

Fully efficient ~8 minimal work units

Multigrid Solvers

• Linear scalar elliptic equations (1970)*	
• Nonlinear • Adaptive grids	FAS
• General boundaries, BC*	
• Discontinuous coefficients	$\nabla(a\nabla u)=f$
• Disordered coefficients	AMG
• Non scalar PDE systems*	elasticity, NS,...
• Non-ellipticity : convection	high Re
• Indefinite : waves	⟷ rays
• Inverse problems	a=? DA
• Time dependent	space + time
• Dense n×n matrices	O(n)
• Crucial features invisible to coarse grids	
• Gauge freedom	
• Topologies	Dirac eqs.
• Near zero modes	

*Rigorous

Compressible Navier-Stokes: 2D

$$-\mu\Delta u + \rho u u_x + \rho v u_y - (\lambda+\mu)(u_x+v_y)_x + p_x = 0$$

$$-\mu\Delta v + \rho u v_x + \rho v v_y - (\lambda+\mu)(u_x+v_y)_y + p_y = 0$$

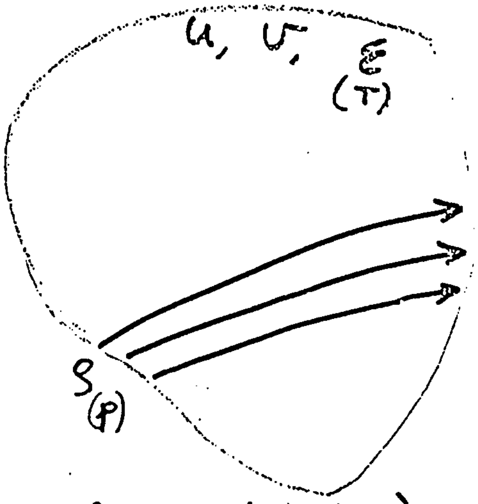
$$-\varkappa\Delta\varepsilon + \rho u\,\varepsilon_x + \rho v\,\varepsilon_y + p(u_x+v_y)$$
$$-\mu(v_x+u_y)^2 - \lambda(u_x+v_y)^2 - 2\mu(u_x^2+v_y^2) = 0$$

$$(\rho u)_x + (\rho v)_y - \nu\Delta\rho = 0$$

$$p = p(\varepsilon,\rho)$$

Non-elliptic BVP.
unknown $u, v, \varepsilon, \rho, p$

B.C.:



Inviscid case: $\lambda, \mu, \varkappa \ll \rho\,\ell \max(|u|,|v|)$

ℓ = length at which (u, v, ε) change. ⇒ Euler eq.

Usually there are viscous layers.

Incomressible: $\begin{pmatrix} Q & 0 & \partial_x \\ 0 & Q & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix}\begin{pmatrix} u \\ v \\ p \end{pmatrix} = 0$

$Q = -\frac{1}{R_e}\Delta + u\partial_x + v\partial_y$.

$\det \uparrow = (-\Delta)Q$ elliptic.

h-principal $\mathbb{L} u = f$ det $\mathbb{L}$

Cauchy-Riemann

$$\begin{pmatrix} \partial_x & \partial_y \\ \partial_y & -\partial_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = 0 \qquad -\Delta$$

2D Stokes

$$\begin{pmatrix} -\Delta & 0 & \partial_x \\ 0 & -\Delta & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = 0 \qquad \Delta^2$$

2D Incompressible Navier-Stokes

$$\begin{pmatrix} Q & 0 & \partial_x \\ 0 & Q & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} \qquad -Q\Delta$$

$$Q = -\frac{1}{R}\Delta + \underline{u} \cdot \underline{\nabla}$$

2D Euler

$$\begin{pmatrix} \rho \underline{u}\cdot\underline{\nabla} & 0 & 0 & 0 & \partial_x \\ 0 & \rho \underline{u}\cdot\underline{\nabla} & 0 & 0 & \partial_x \\ \rho\partial_x & \rho\partial_y & \underline{u}\cdot\underline{\nabla} & 0 & 0 \\ p\partial_x & p\partial_y & 0 & \rho \underline{u}\cdot\underline{\nabla} & 0 \\ 0 & 0 & -P_\rho & -P_\varepsilon & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ \rho \\ \varepsilon \\ p \end{pmatrix} \qquad \rho^3 (\underline{u}\cdot\underline{\nabla})^2 \times \left[(\underline{u}\cdot\underline{\nabla})^2 - a^2\Delta\right]$$

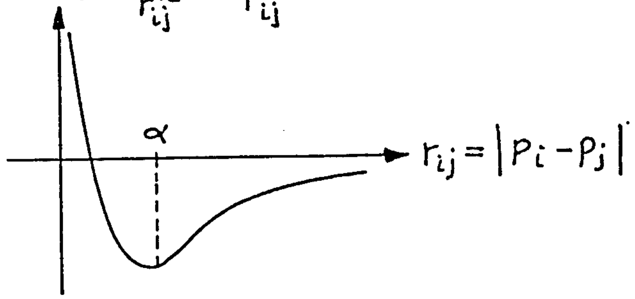
$$a^2 = P_\rho + \frac{p}{\rho^2} P_\varepsilon$$

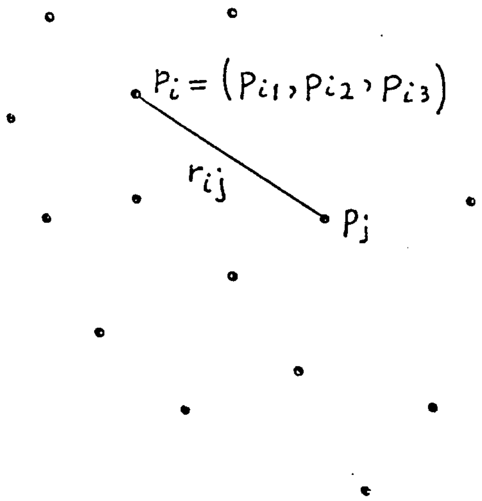
Compressible Navier-Stokes (on the viscous scale) $\kappa\mu(2\mu+\lambda)\Delta^3(\underline{u}\cdot\underline{\nabla})$

Central Cauchy-Riemann Δ^{2h}

Central (Navier-) Stokes $Q^h \Delta^{2h}$

$$V_{ij} = \frac{a_{ij}}{r_{ij}^{12}} - \frac{b_{ij}}{r_{ij}^{6}}$$





$$E^{\circ}(p) = E^{\circ}(p_1, p_2, \dots) = \sum_{i \neq j} V_{ij}(|p_i - p_j|)$$

$$E^{\circ}(p^{\circ}) = \min_{p} E^{\circ}(p)$$

<u>With external forces</u>: $E(p) = E^{\circ}(p) - \sum_i f_i \cdot p_i$

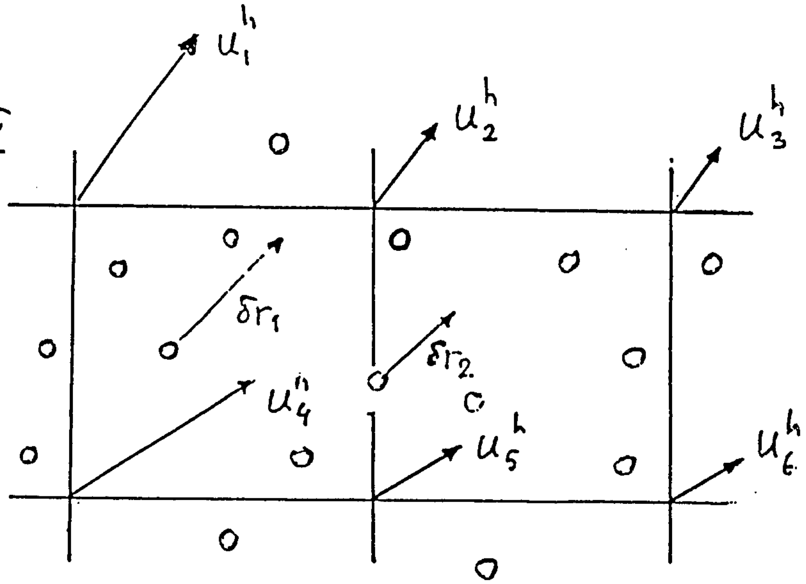
<u>Find</u> p^* such that $E(p^*) = \min E(p)$

Particle - by - particle minimization

$\Longrightarrow$ Extreme slow down

Needed : collective moves

Displacement field u^h



δr_i = Interpolation from $u^h = (u_1^h, u_2^h, \ldots)$

$\Rightarrow E(r+\delta r) = E^h(u^h)$ explicitly (for smooth u^h)

- Minimize $E^h(u^h)$, only <u>then</u> displace particles

$$E(r) = -\sum f^i \cdot r_i + \sum_{i,j} V_{ij}\left(|r_i - r_j|\right) \qquad + \sum g^i r_i^2$$

$$\delta r_i = \sum_k \lambda_i^k u_k \, , \quad \sum_k \lambda_i^k = 1 \, , \quad u = u^h$$

$$\delta r_j = \sum_k \lambda_j^k u_k \, , \quad \sum_k \lambda_j^k = 1$$

$$\delta(r_i - r_j) = \sum_k (\lambda_i^k - \lambda_j^k)\, u_k \qquad \sum_k (\lambda_i^k - \lambda_j^k) = 0$$

$$= \sum_{k,\ell} \mu_{ij}^{k\ell}\, (u_k - u_\ell)$$

$$V_{ij}(|r_i - r_j|) = V_{ij}\left(\left|r_i^o - r_j^o + \sum \mu_{ij}^{k\ell}\,(u_k - u_\ell)\right|\right)$$

$$\approx V_{ij}(|r_i^o - r_j^o|) + a_{ij}^1 \sum \mu_{ij}^{k\ell}\,(u_k - u_\ell)$$

$$+ a_{ij}^2 \left[\sum \mu_{ij}^{k\ell}\,(u_k - u_\ell)\right]^2 + \dots$$

for smooth u^h : $|u_k^h - u_\ell^h| \leq \varepsilon_{k\ell}$ for k near ℓ

$$E^h(u^h) = \sum f_k^h u_k^h + \sum b_{k\ell}^h u_k^h u_\ell^h + \dots$$

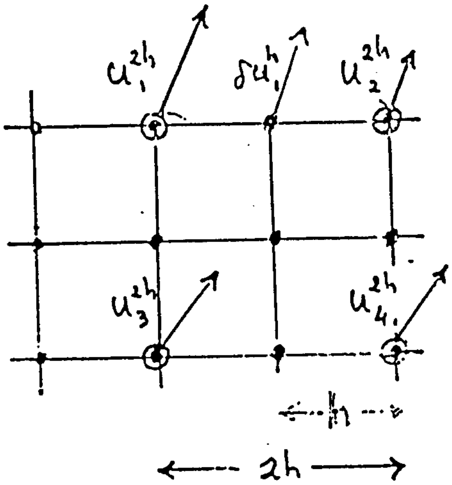
$\Longrightarrow$ Similarly on coarser grids

$\Longrightarrow$ Elasticity - Plasticity - Strain limits

Minimization of $E^h(u^h)$

Point-by-point minimization

$\longrightarrow$ slow down!



- Transfer to grid 2h

δu_i^h = Interpolation from u^{2h}

Construct explicitly

$$E^{2h}(u^{2h}) = E^h(u^h + \delta u^h(u^{2h}))$$

- Etc. $E^{4h}(u^{4h})$, $E^{8h}(u^{8h})$, ...

Each u^{2H} describes smooth change of u^H

- Similarly if the original problem is a grid problem

Fine scale	Coarse scale
Uniformly elliptic eq.	Same eq.
Wave eq. $\Delta u + k^2 u = 0$	"Rays" eq.
Particles Lennard-Jones energy minimization	Continuum eqs. Elasticity eqs.

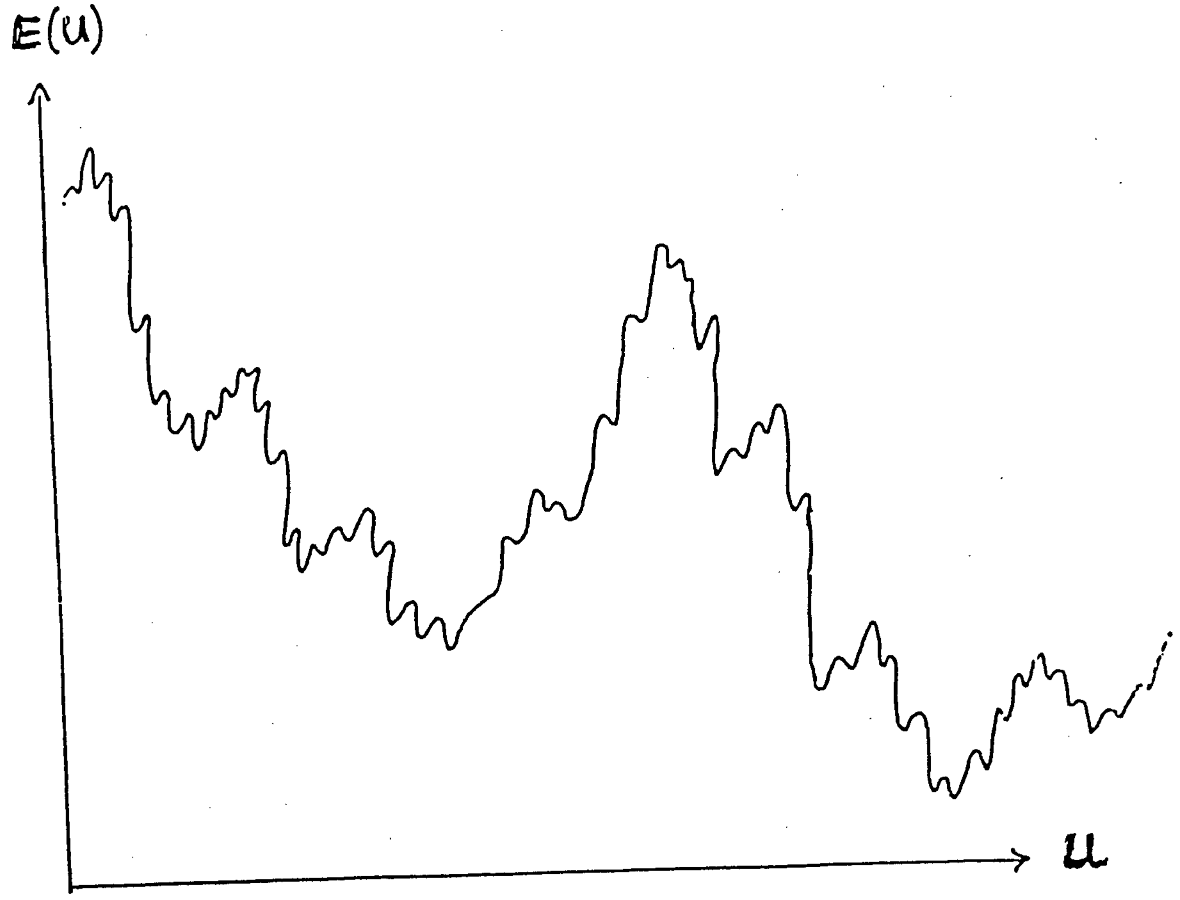
Derivation of macroscopic "equations"

(simpler than renormalization group)

Stochastic	$\longleftrightarrow$	Deterministic
Deterministic		Stochastic

Optimization $\min E(u)$

- Fast convergence near optimum
- Global optimization
  Escape false attraction basins



multi-scale attraction basins

min $E(u)$

False attraction basins

Particle by particle minimization
trapped in local attraction basins.

Simulated annealing $\qquad P(u) \sim e^{-E(u)/T}$
$\qquad T \to 0$

trapped in large-scale attraction basins.

Multi-level annealing
Collective moves, at all scales
converges (fast, in probability).

- A large-scale move is decided only after optimizing around it at all finer scales.
- Fast annealing at each level.
- Recombinations at each level.

Thermal Statistical Fluctuations

1. Need to produce (stochastically, in a correct probability distribution) independent configurations
2. Need to produce very many of them (to average out deviations)

Multigrid Monte-Carlo

1. Collective moves, at all scales
2. Producing many samples on coarse grids (little work per sample, averaging out large-scale fluctuations; local fluctuations are self-averaging)

Optimal:
Thermodynamic limit $\pm \varepsilon$
in $O(\varepsilon^{-2})$ computer operations

- General non-linear PDE

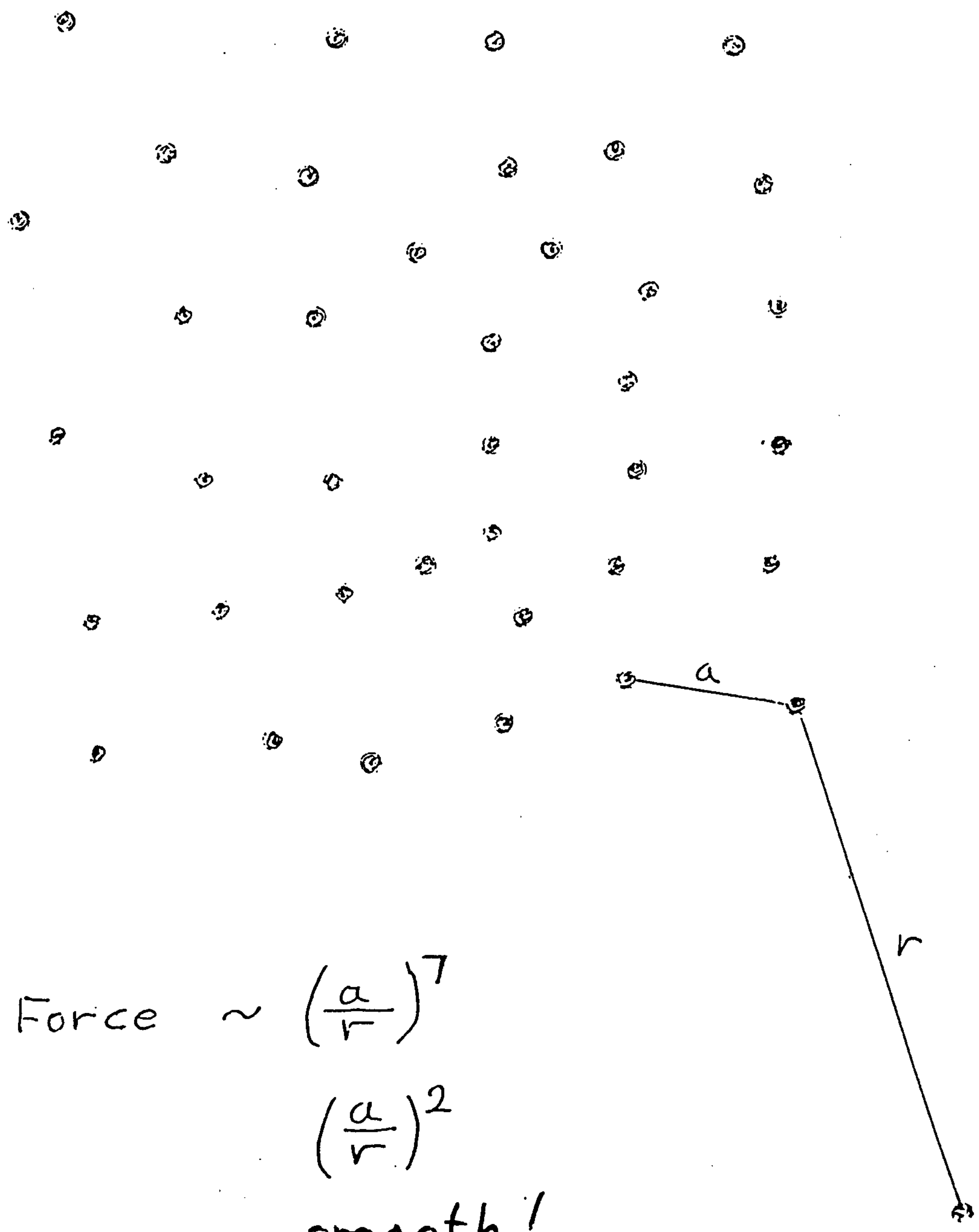
  steady state: Elliptic, non-elliptic
  time dependent
  inverse problems. Optimal control
  waves ⟷ rays. Many-eigenfunction.
  topological singularites + disorder (Dirac eqs.)

- Integral transform, equations
- n-body interactions
- Molecular mechanics : Ground states
  Equilibrium
  Dynamics
- Global optimization
- Linear Programming
- Monte Carlo
- Determinant
- Multiple-fermion path integrals
- Derivation of macroscopic equations
- Image processing
- Tomography   X rays, NMR, PET, ... radar

A. Brandt

Decomposition of Forces and Fast Force Summation

Charges



$$\text{Force} \sim \left(\frac{a}{r}\right)^7$$

$$\left(\frac{a}{r}\right)^2$$

smooth!

Integral Transforms

$$V(x) = \int G(x,y)\, u(y)\, dy$$

$$x = (x_1, x_2, x_3), \qquad y = (y_1, y_2, y_3)$$

Asymptotically **smooth kernels**

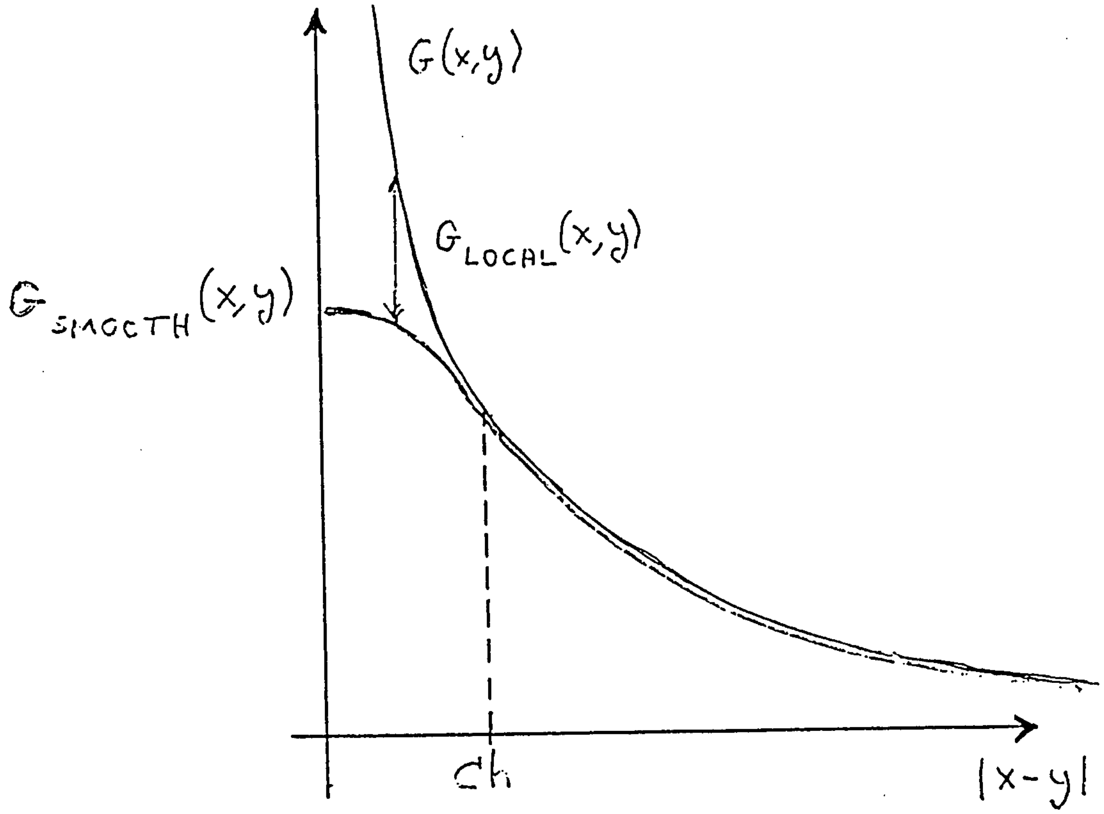
$G(x,y) =$	$e^{i\,x\cdot y}$	Fourier
	$e^{-x\cdot y}$	Laplace
	$e^{-(x-y)^2/\sigma^2}$	Gauss
	$\lvert x-y\rvert^{-k}, \ \log\lvert x-y\rvert$	Potential

Oscillatory kernels

$$\tilde{G}(x,y)\, e^{ik|x-y|}$$

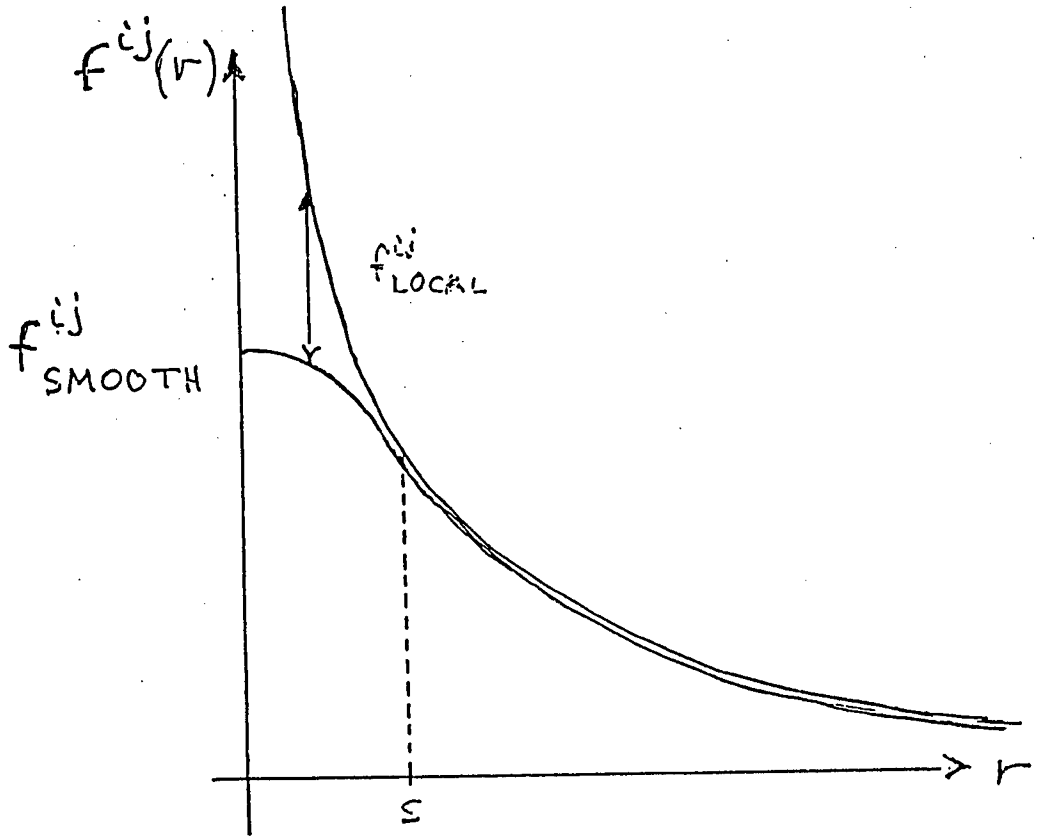
Integro-differential equations

$$V(x_i) = \sum_{y_j \in \Omega^h} G(x_i, y_j)\, u^h(y_j) \qquad x_i \in \tilde{\Omega}^h$$



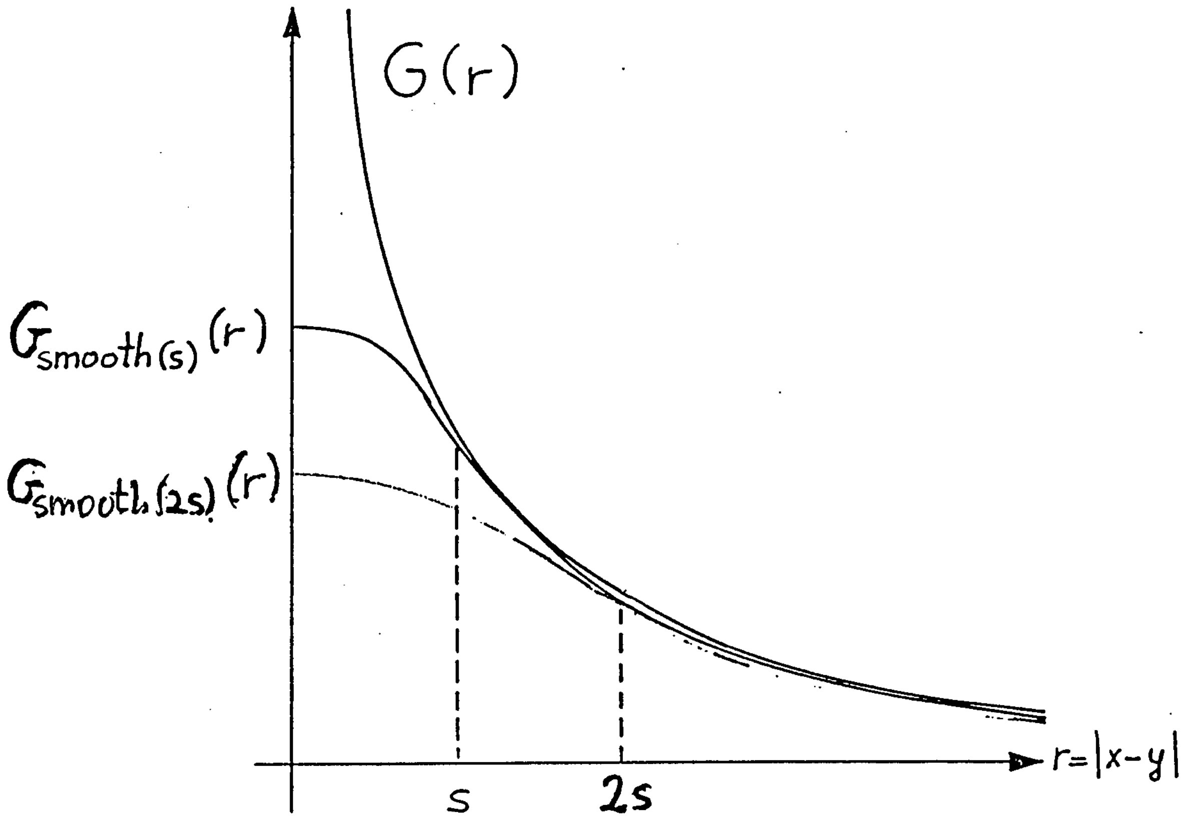
$$G(x,y) = G_{LOCAL}(x,y) + G_{SMOOTH}(x,y)$$

directly coarser grid



$$f^{ij}_{LOCAL}(r) = 0 \quad \text{for} \quad r \geqslant s \quad \sim a$$

$$\left| \frac{\partial^k}{\partial r^k} f^{ij}_{SMOOTH}(r) \right| \leqslant C s^{-k} \left| f^{ij}_{SMOOTH}(r) \right|$$



$$\left| \frac{\partial^k}{\partial r^k} G_{smooth(s)}(r) \right| \leqslant \frac{C(s)}{s^k}$$

$$v(x_i) = \sum_{j=1}^{m} G(x_i, y_j)\, u(y_j), \qquad (i = 1, \dots, n)$$

$\left.\begin{matrix} x_1, \dots, x_n \\ y_1, \dots, y_m \end{matrix}\right\}$ particles or gridpoints

$\left.\begin{matrix} X_1, \cdots X_N \\ Y_1, \cdots Y_M \end{matrix}\right\}$ (coarser) uniform grids
meshsize h

p-order interpolation

$$f(x_i) = \sum_k \omega_{ik} f(X_k) + O\left((\gamma h)^p f^{(p)}\right)$$

$$\tilde{f}(y_j) = \sum_\ell \tilde{\omega}_{j\ell} \tilde{f}(Y_\ell) + O\left((\gamma h)^p \tilde{f}^{(p)}\right)$$

$$v(x_i) \approx \sum_{j,\ell} \tilde{\omega}_{j\ell}\, G(x_i, Y_\ell)\, u(y_j)$$

$$= \sum_\ell G(x_i, Y_\ell)\, U(Y_\ell) \qquad U(Y_\ell) = \sum_j \tilde{\omega}_{j\ell}\, u(y_j)$$

① anterpolation

$$\approx \sum_{\ell,k} \omega_{ik}\, G(X_k, Y_\ell)\, U(Y_\ell)$$

$$= \sum_k \omega_{ik}\, V(X_k) \qquad V(X_k) = \sum_k G(X_k, Y_\ell)\, U(Y_\ell)$$

③ interpolation ② recursion

④ **correction:** $v(x_i) = v(x_i) + \sum G_{local}(x_i, y_j)\, u(y_j)$

n-particle forces

Calculated directly: $9n^2$ operations

Multiscale calculation: Cn operations

C (dimension, accuracy (def.), uniformity)

Only the local part of the forces should be used/updated in local motions

Smooth part of the force
— in larger-scale (collective) motions.

operations: $O(Ns^{d+1})$

N gridpoints

$O(h^s)$ accuracy

d dimension

Coefficient $\ll 1$

Harmonic Kernels: $\sim 6dsN$

Via multigrid Poisson solver

Integro-Differential Equation

$$\underset{\substack{\uparrow \\ \text{differential}}}{L}u(x) = \int G(x,y)\,u(y)\,dy$$

Multigrid Solver

distributive relaxation:

1st order	$+\delta$	$-\delta$	
2nd order	$+\delta$	-2δ	$+\delta$

Solution cost $\approx$ <u>one</u> fast transform
(one fast evaluation of the
discretized integral transform)

Elements of linear multigrid (MG) for solving elliptic PDEs

Klaus Stüben

GMD/SCAI
Schloß Birlinghoven
D-53757 St. Augustin
Germany

Contents:

1. General remarks, history
2. Literature
3. Terminology
4. Basic principles
   - Smoothing
   - Two-grid method
   - Multigrid method
   - Computational work
5. Simple examples

General remarks

What is multigrid NOT?

A particular solver

What IS multigrid?

A general strategy for constructing solvers

This strategy exploits the fact that a problem can be approximated on different scales of resolution ("grids", "levels").

General idea

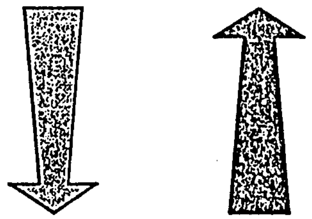
Compute only those "parts" of a solution on a fine fine "grid" which REQUIRE a fine resolution.

Compute other "parts" on coarser "grids"!

Features of multigrid

(1) Provides "optimal" methods (i.e.: O(N))
(2) General applicability
(3) Involves only "local processes"
(Parallel computer architectures!)

Classical methods

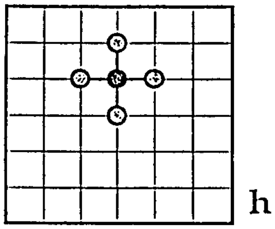


increasing efficiency

increasing generality

Gauß-Elimination
Gauß-Seidel
SOR
ADI
Buneman

Comparison with MG



Poisson equation:

symmetric differencing (2nd order)
unit square: $h=1/n$
# unknowns: $N = n^2$

Solution up to discretization error $\varepsilon = O(h^2)$

method	# operations	time, h=1/256
Gauss-Elimin.	$O(N^2)$	approx. 1 day
SOR	$O(N^{3/2}) \log \varepsilon$	approx. 30 min
ADI	$O(N \log N) \log \varepsilon$	4 min
Buneman	$O(N \log N)$	15.5 sec
MG (iterative)	$O(N) \log \varepsilon$	18.5 sec
MG (FMG)	$O(N)$	7.6 sec

$\log \varepsilon = O(\log N)$

MG1-3

Range of MG applicability

General domains
General boundary conditions
Variable coefficients
Linear / non-linear problems
Scalar problems / systems of PDEs
Singular / nearly singular problems
Eigenvalue problems
⋮

Local refinements
Local coordinates

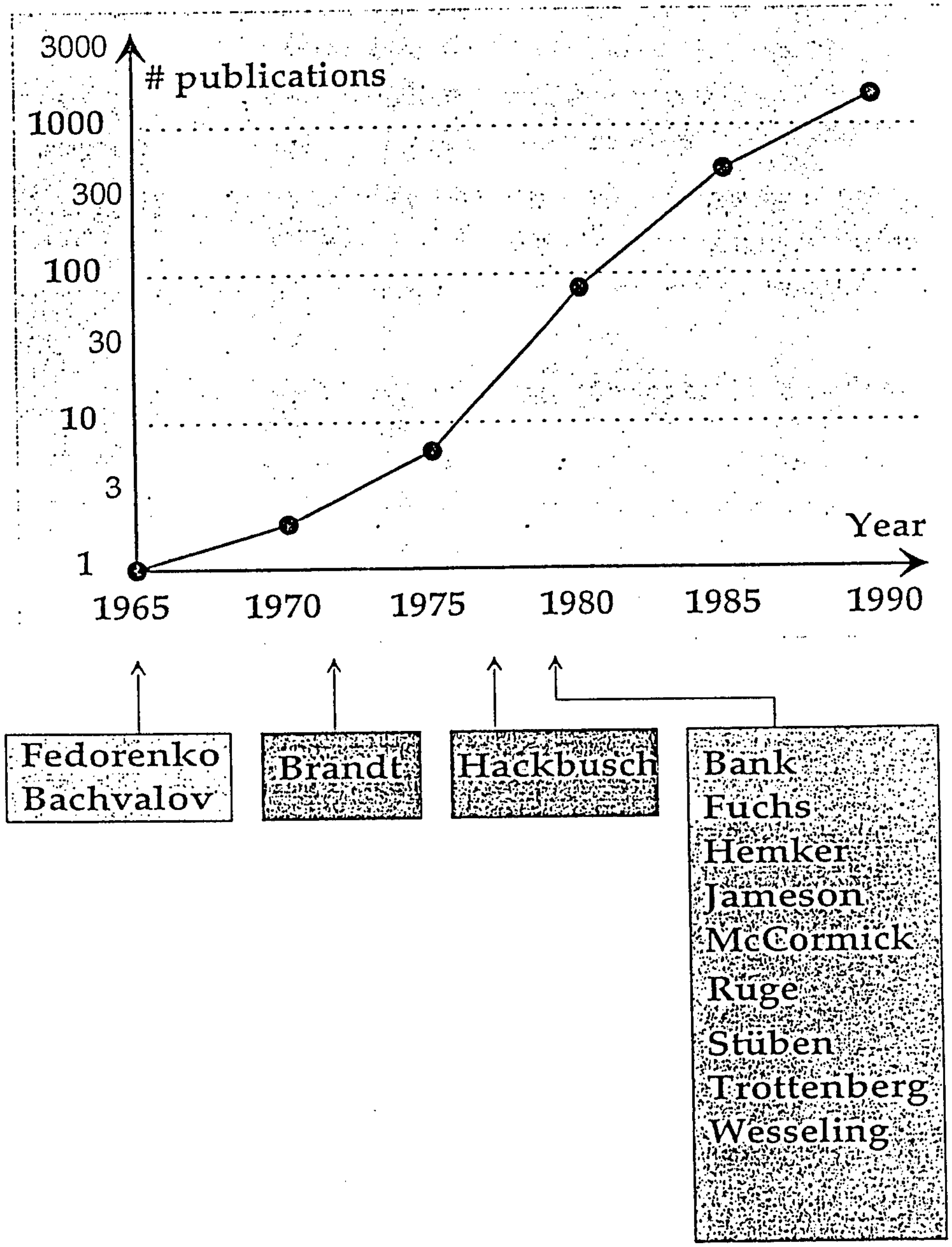
Finite differences
Finite volumes
Finite elements

However, ...

... the development of concrete multigrid methods
for complex problems may be non-trivial:

The "components" of a multigrid method have to be adjusted to particular situations.

Historical development



Fedorenko
Bachvalov

Brandt

Hackbusch

Bank
Fuchs
Hemker
Jameson
McCormick
Ruge
Stüben
Trottenberg
Wesseling

Pioneering papers

Discovery of multigrid (theoretical)

Fedorenko, R.P.: *The speed of convergence of an iterative process,* USSR Comput. Math. and Math. Phys. 4,3 (1964)

Bakhvalov, N.S.: *On the convergence of a relaxation method with natural constraints on the elliptic operator,* USSR Comput. Math. and Math. Phys. 6,5 (1966)

Papers marking the "beginning" of multigrid

Brandt, A.: *Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems,* Lecture Notes in Physics 18, Springer (1973)

Brandt, A.: *Multi-level adaptive solutions to boundary-value problems,* Math. Comp. 31 (1977)

Re-discovery of multigrid (theoretical)

Hackbusch, W.: *On the multigrid method applied to difference equations,* Computing 20 (1978)

Introductions to MG

Stüben, K.; Trottenberg, U.: *Multigrid methods: Fundamental algorithms, model problem analysis and applications,* Lecture Notes in Mathematics 960, Springer (1982)

Br. ndt, A.: *Multigrid techniques: 1984 Guide with applications to fluid dynamics,* GMD-Studie No. 85 (1984)

Hackbusch, W.: *Multigrid methods and applications,* Springer Series in Comp. Math. 4, Springer (1985)

McCormick, S. (ed.): *Multigrid methods,* Frontiers in Applied Mathematics, Vol. 5, SIAM, Philadelphia (1987)

Briggs, W.: *A multigrid tutorial,* SIAM, Philadelphia (1987)

Wesseling, P.: *An introduction to multigrid methods,* Pure and Applied Mathematics series, John Wiley and Sons (1992)

Joppich, W.; Mijalkovic, S.: *Multigrid Methods for Process Simulation,* Computational Microelectronics 17, Selberherr (ed.), Springer Verlag, 1993

Proceedings

Copper Mountain MG Conference

1st: Appl. Math. Comp. 13 (1983)
2nd: Appl. Math. Comp. 19 (1986)
3rd: Lecture Notes in Pure and Appl. Math. 110,
Marcel Dekkar (1988)
4th: SIAM, Philadelphia (1989)
5th: Communications in Applied Numerical Methods;
Special Issue on MG, Vol 8, No 9&10,
John Wiley & Sons, New York (1992)
6th: NASA Conference Publication 3224, Parts 1&2,
NASA, Hampton, Virginia (1993)
7th: ??

European MG Conference

1st: Lecture Notes in Mathematics 960, Springer (1982)
2nd: Lecture Notes in Mathematics 1228, Springer (1986)
3rd: International Series of Numerical Mathematics, Vol. 98,
Birkhäuser (1991)
4th: International Series of Numerical Mathematics, Vol. 116,
Birkhäuser (1994)

The three basic principles of multigrid

Relaxation

Coarse-grid correction

Nested iteration

None of these principles yields an efficient method by itself!

First step:

Multigrid iteration ("cycling")

Relaxation + Coarse-grid correction

Second step:

Full multigrid (FMG)

Multigrid cycling + Nested iteration

Relaxation as a solver

Basic principle:

Given: $A X = b$ Splitting: $A = A_1 + A_2$

One iteration step $x \rightarrow \bar{x}$:

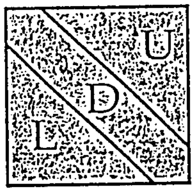
$$A_1 \bar{x} + A_2 x = b$$

$$\Longrightarrow \quad \bar{x} = -A_1^{-1} A_2 x + A_1^{-1} b$$

$\rho(-A_1^{-1} A_2)$ = asympt. conv. factor (spectral radius)

Typical PDE situation

A =



Jacobi relaxation:

$A_1 = D, \; A_2 = L + U$

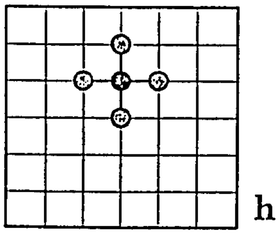
Gauss-Seidel relaxation:

$A_1 = D + L, \; A_2 = U$

$\rho = 1 - O(h^2)$ <u>h - dependent!</u>

Relaxation methods are very inefficient solvers!

Relaxation as a smoother



Poisson equation

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h U^h = f^h$$

Pointwise Gauss-Seidel relaxation

For all grid points do

$$u_{i,j}^h = \frac{1}{4}\left[h^2 f_{i,j}^h + u_{i-1,j}^h + u_{i+1,j}^h + u_{i,j-1}^h + u_{i,j+1}^h \right]$$

Effect on *error* $v^h = U^h - u^h$

$$v_{i,j}^h = \frac{1}{4}\left[v_{i-1,j}^h + v_{i+1,j}^h + v_{i,j-1}^h + v_{i,j+1}^h \right]$$

→ averaging process

The *error* gets *smooth* very quickly!

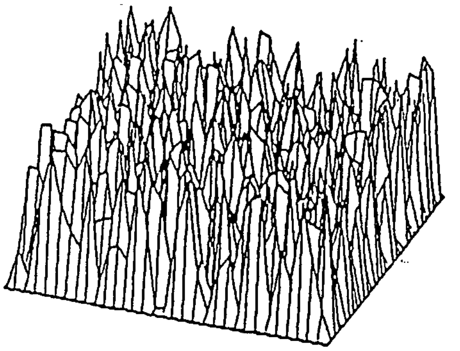
Relaxation processes are local:

Locally, information is spread quickly

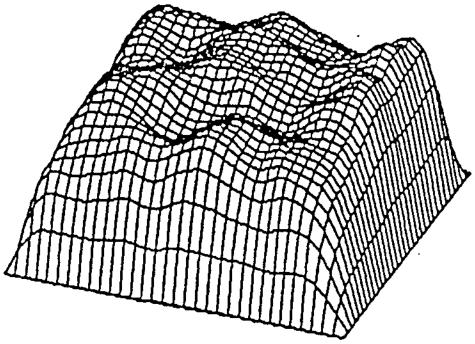
Globally, information is spread slowly (in particular, boundary values!)

Influence of (pointwise) Gauss-Seidel relaxation on the error

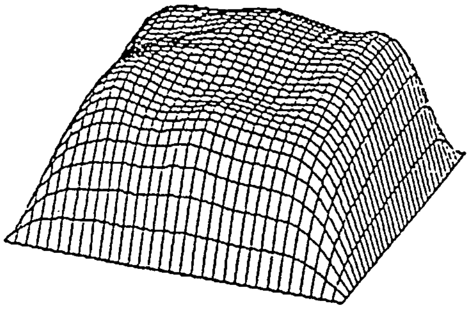
Poisson equation, *uniform grid*



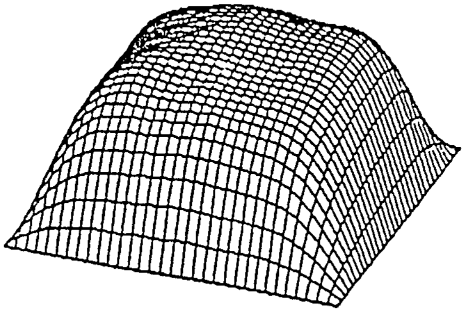
Error of initial guess



Error after 5 relaxations



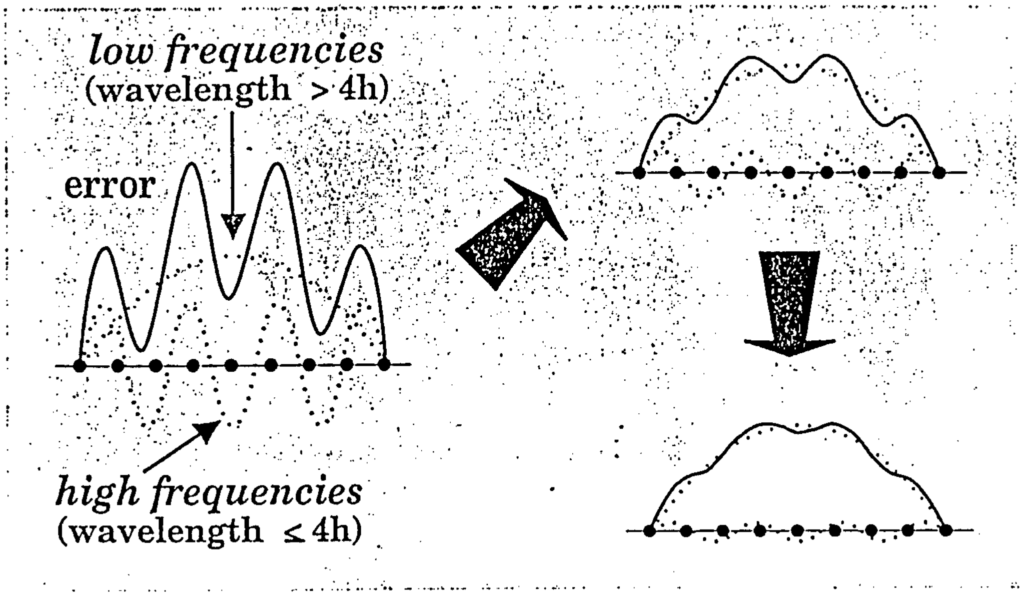
Error after 10 relaxations



Error after 15 relaxations

Smoothing analysis: Fourier decomposition

Fourier decomposition of *error* (1D)



High frequency reduction factor per sweep
("Smoothing factor" μ)

Independent of h!

Analysis by (local) Fourier analysis
Poisson equation: GS: μ=0.5, RB: μ=0.25

Efficient smoothers exist for all "sufficiently elliptic" PDEs (or systems of PDEs)

Typical reduction: 1 order of magnitude in 2-4 sweeps

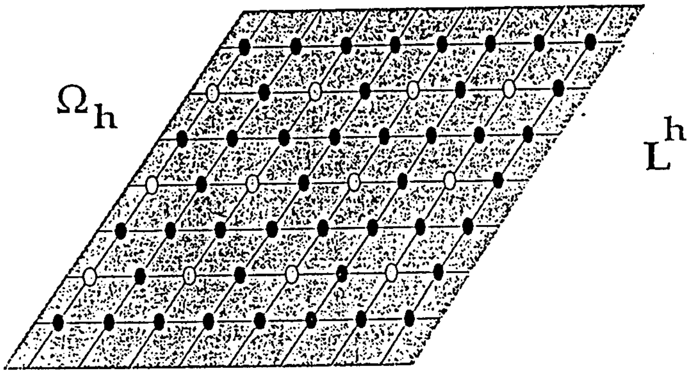
Hierarchy of discretizations

fine grid: Ω_h coarser grid: Ω_H

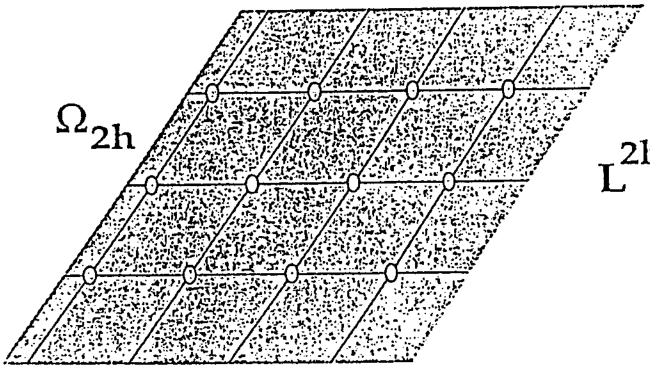
$L^h U^h = f^h$ difference equations on Ω_h

$L^H U^H = f^H$ difference equations on Ω_H

Standard coarsening (H=2h):



$$L^h = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h$$



$$L^{2h} = \frac{1}{4h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_{2h}$$

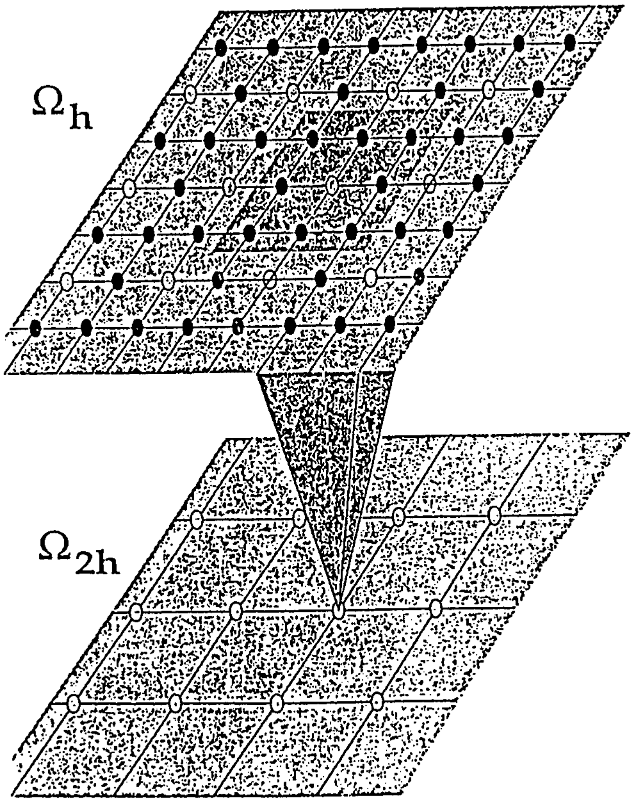
Data transfer between different grids

fine grid: Ω_h coarser grid: Ω_H

I_h^H transfer from Ω_h to Ω_H: *restriction*

I_H^h transfer from Ω_H to Ω_h: *prolongation*

Standard coarsening (H=2h):



Restriction:

("full weighting")

$$I_h^{2h} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Prolongation:

(linear interpolation)

$$I_{2h}^h = \frac{1}{4}\left]\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}\right[$$

Coarse-grid correction

Given: $L^h U^h = f^h$ approximate solution: u^h

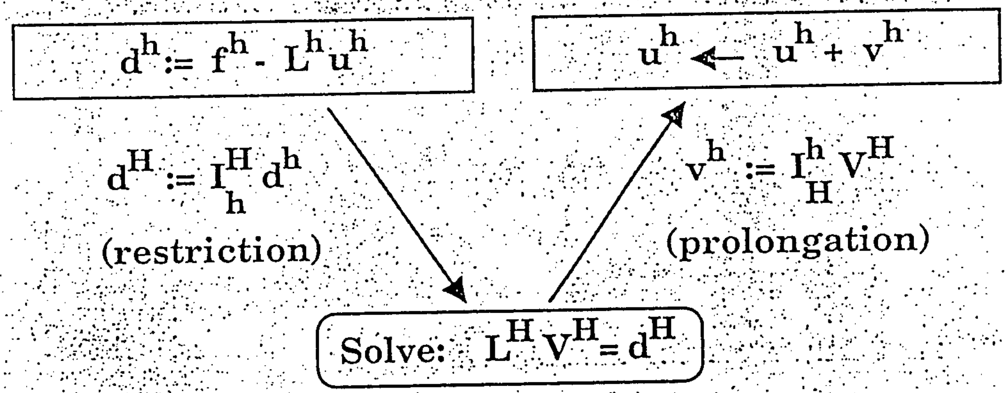
Correction equation:

$L^h V^h = d^h$ with $d^h := f^h - L^h u^h$ *(= residual)*

$\Longrightarrow U^h = u^h + V^h$

Given in addition: Ω_H and L^H

Coarse-grid correction:



$d^h := f^h - L^h u^h$

$u^h \leftarrow u^h + v^h$

$d^H := I_h^H d^h$

(restriction)

$v^h := I_H^h V^H$

(prolongation)

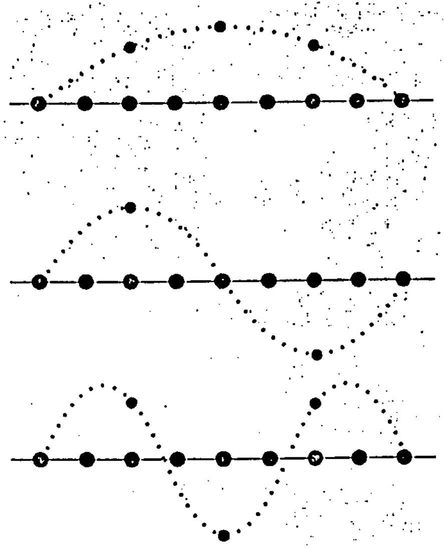
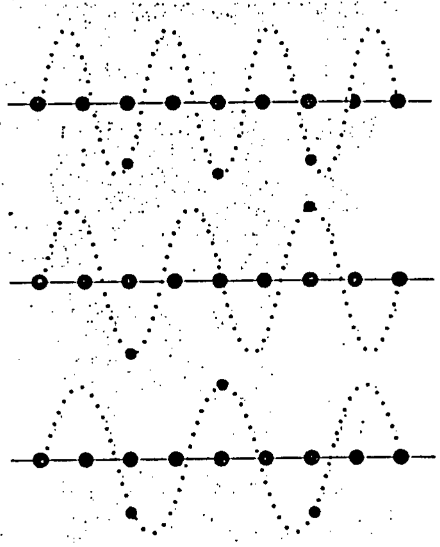
Solve: $L^H V^H = d^H$

This process, applied iteratively, CANNOT converge!

$I_h^H d^h = 0 \Longrightarrow$ no correction!

Coarse-grid aliasing of Fourier components

Assume: Fourier decomposition of *error*

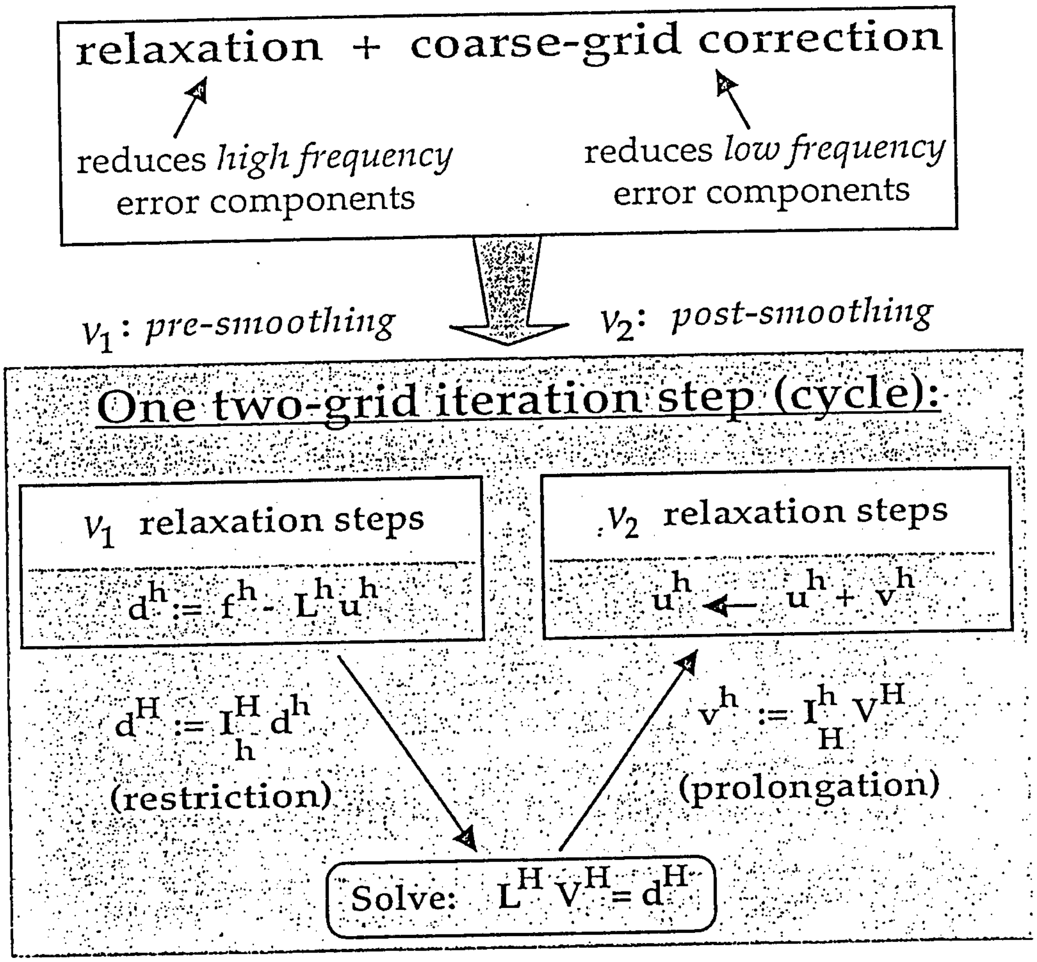
Fourier frequencies (1D)	
low frequencies (wavelength > 4h) Visible on coarse grid!	*high frequencies* (wavelength ≤ 4h) Invisible on coarse grid!
	

High frequency Fourier components CANNOT be corrected by coarse grid!

However:

Coarse-grid correction makes sense, if *low fequencies* are dominating in the *error!*

The (iterative) two-grid cycle



relaxation + coarse-grid correction

reduces *high frequency* error components

reduces *low frequency* error components

ν_1: *pre-smoothing*

ν_2: *post-smoothing*

One two-grid iteration step (cycle):

ν_1 relaxation steps

$d^h := f^h - L^h u^h$

ν_2 relaxation steps

$u^h \leftarrow u^h + v^h$

$d^H := I_h^H d^h$

(restriction)

$v^h := I_H^h V^H$

(prolongation)

Solve: $L^H V^H = d^H$

Convergence independent of h!

ν_1, ν_2 small (typically 1 or 2)

Rough convergence prediction: smoothing factor

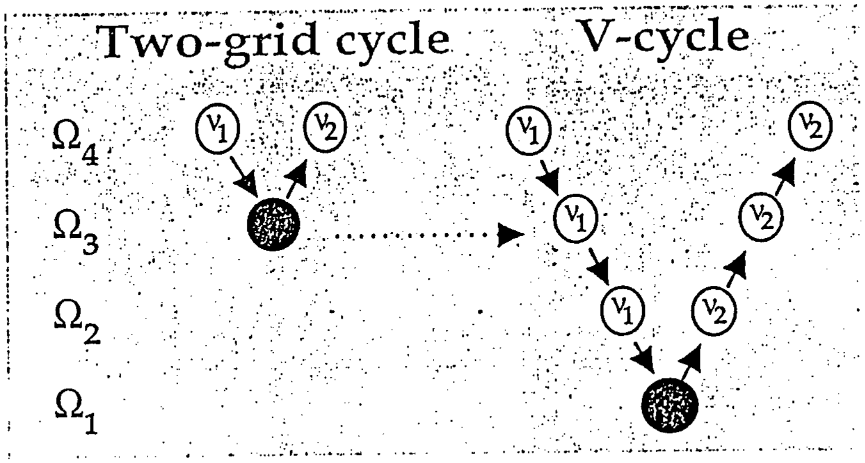
Analysis by (local) Fourier analysis

<u>However:</u> Two-grid method is not yet efficient!

Typical multigrid cycles

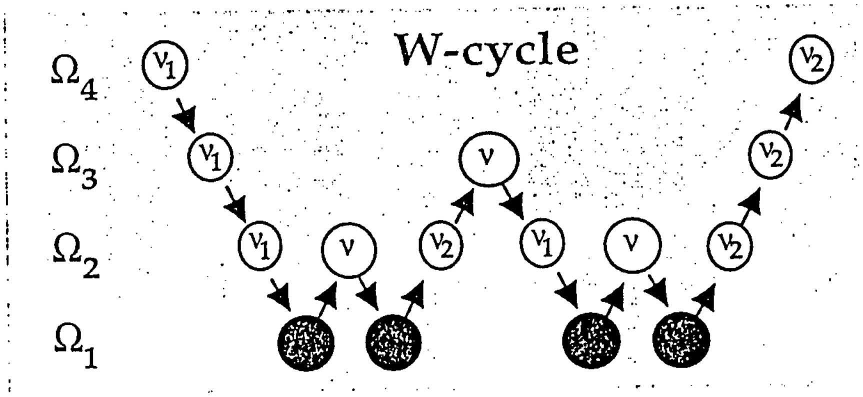
Notation:

○	ν smoothing steps	↘	restriction
●	solve grid equation	↗	prolongation



Cycle index:

$\gamma=1$



$\gamma=2$

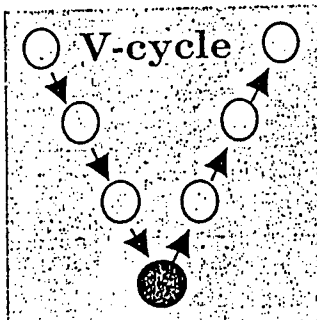
<u>Notation:</u> $V(\nu_1, \nu_2)$ - cycle, $W(\nu_1, \nu_2)$ - cycle

<u>Compromise:</u> F-cycle

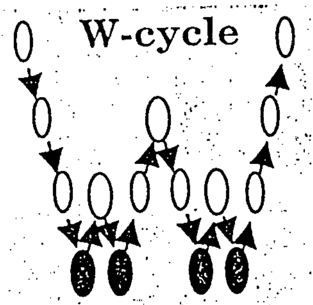
Work count

$\mathcal{W}$:= 1 relaxation sweep on finest grid

$\nu = \nu_1 + \nu_2$,   standard coarsening



	2D	3D
	$\nu \mathcal{W}$	$\nu \mathcal{W}$
	$\nu \mathcal{W}$ /4	$\nu \mathcal{W}$ /8
	$\nu \mathcal{W}$ /16	$\nu \mathcal{W}$ /64
	⋮	⋮
	4/3 $\nu \mathcal{W}$	8/7 $\nu \mathcal{W}$
$\nu = 3$:	4 $\mathcal{W}$	3.5 $\mathcal{W}$



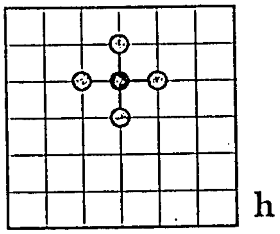
	2D	3D
	$\nu \mathcal{W}$	$\nu \mathcal{W}$
	$\nu \mathcal{W}$ /2	$\nu \mathcal{W}$ /4
	$\nu \mathcal{W}$ /4	$\nu \mathcal{W}$ /16
	⋮	⋮
	2 $\nu \mathcal{W}$	4/3 $\nu \mathcal{W}$
$\nu = 3$:	6 $\mathcal{W}$	4 $\mathcal{W}$

Typical: Error reduction by 1 order of magnitude with work equivalent to 5-10 $\mathcal{W}$

Solution up to discretization error ε

$O(N) \log \varepsilon = O(N) \log N$

Standard MG for Poisson's equation



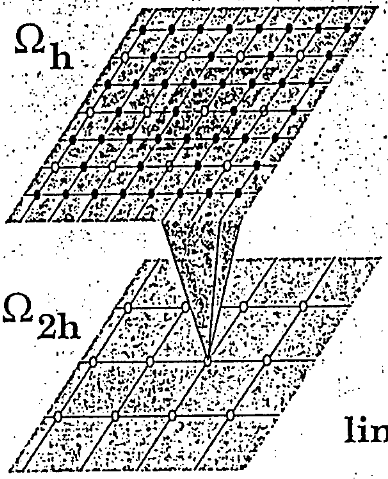
Poisson equation

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h U^h = f^h$$

Relaxation: GS

L^h

standard (h)



I_h^{2h}

full weighting

L^{2h}

standard (2h)

I_{2h}^h

linear interpolation

Convergence factors (h=1/128, 7 grid levels)

ν_1, ν_2	μ^ν	W-cycle	V-cycle
1, 0	0.500	0.395	0.400
1, 1	0.250	0.190	0.190
2, 1	0.125	0.117	0.120

Summary on MG components

Discretization	MG straightforward if "sufficiently elliptic"
Relaxation	Gauss-Seidel type (point-, line-, blockwise); collective and distributive relaxations; ILU-iterations

Coarsening strategy	Standard coarsening (h,2h,4h,...);
	Semi- or adaptive coarsening;
	Multi-coarsening; Variational coarsening

Coarse-grid operators	Analogous on all grids; Galerkin operators: $L^H = I_h^H L^h I_H^h$
Prolongation	Linear interpolation; "Adaptive" interpolation (e.g. discrete eqns.)

Restriction	Local averaging (e.g. full weighting);
	Transpose of prolongation;
	"Adaptive" weighting

Cycle-Typ	V-, F-, W-cycle; "Adaptive" cycles
ν_1, ν_2	Typically 1 or 2, usually $\nu_1 \geq \nu_2$; dynamically chosen

Relevant for non-PDE problems:
Variational coarsening

Optimized MG for Poisson's equation

GS-relaxation + straight injection

$$I_h^{2h} = \frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} \approx \frac{1}{16}\begin{bmatrix}0&0&0\\0&16&0\\0&0&0\end{bmatrix} = \begin{bmatrix}1\end{bmatrix}$$

Residuals do not change much locally

Convergence factors

ν_1, ν_2	μ^ν	W-cycle	V-cycle
1, 0	0.500	0.440	*div*
1, 1	0.250	0.197	0.199
2, 1	0.125	0.088	0.089

RB-relaxation + half injection

$$I_h^{2h} = \frac{1}{16}\begin{bmatrix}\boxed{1}&2&\boxed{1}\\2&\boxed{4}&2\\\boxed{1}&2&\boxed{1}\end{bmatrix} \approx \frac{1}{16}\begin{bmatrix}\boxed{0}&2&\boxed{0}\\2&\boxed{8}&2\\\boxed{0}&2&\boxed{0}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\end{bmatrix}$$

Residuals vanish *at black points*

Residuals do not change much locally *at the red points*

Convergence factors

ν_1, ν_2	μ^ν	W-cycle	V-cycle
1, 0	0.250	0.487	0.809
1, 1	0.063	0.118	0.173
2, 1	0.016	0.034	0.059

Variational Coarsening

$$AU = b \quad \text{on grid } h, \; A \text{ symm. pos. def.}$$

$$\Updownarrow$$

$$\tfrac{1}{2} U^T A U - U^T b = \text{min!}$$

$I_{2h}^h V^{2h}$ coarse-grid correction to u: $\quad U \approx u + I_{2h}^h V^{2h}$

$$\Downarrow$$

minimize

$$\tfrac{1}{2}\left(u + I_{2h}^h V^{2h}\right)^T A \left(u + I_{2h}^h V^{2h}\right) - \left(u + I_{2h}^h V^{2h}\right)^T b$$

$$\tfrac{1}{2} V^{2hT} \underline{I_{2h}^{hT} A I_{2h}^h} V^{2h} - V^{2hT} \underline{I_{2h}^{hT}(b - Au)}$$

$$\Updownarrow$$

$$A^{2h} V^{2h} = R^{2h} \quad \text{where}$$

$$A^{2h} = I_{2h}^{hT} A I_{2h}^h, \quad R^{2h} = I_{2h}^{hT}(b - Au)$$

Symmetric, Conservative — Full weighting

Variational coarsening

Given: L^h, Ω_h, Ω_H and I_H^h

Define: $I_h^H := (I_H^h)^T$ and $L^H := I_h^H L^h I_H^h$

Galerkin operator

L^h symmetric ⟹ L^H symmetric

L^h pos. definite ⟹ L^H pos. definite

Variational principle (L^h pos. definite)

$v^h := U^h - u^h$ error *before* coarse-grid correction

$\overline{v}^h := U^h - \overline{u}^h$ error *after* coarse-grid correction

Then: $\|\overline{v}^h\| = \min_{\delta^h \in \mathcal{R}} \|v^h + \delta^h\|$

$\|\cdot\|$ energy norm $\mathcal{R} := \text{range}(I_H^h)$

Remarks:

Applicable to non-PDE problems

Only applicable for linear problems

The computation of L^H is relatively expensive

Origin: finite elements

Proof of the variational principle

$< u^h, w^h >_h := (L^h u^h, w^h)$ *energy inner product*

$\mathcal{R} :=$ range(I_H^h), $\mathcal{R}^\perp :=$ orthogonal complement

Two-level correction operator: $K^h = I^h - I_H^h (L^H)^{-1} I_h^H L^h$

Properties:

K^h is symmetric w.r.t. $< . , . >$

$v^h \in \mathcal{R} \longrightarrow K^h v^h = 0$

$v^h \in \mathcal{R}^\perp \longrightarrow K^h v^h = v^h$

$\longrightarrow$ K^h is orthogonal projector onto $\mathcal{R}^\perp$

$I^h - K^h$ is orthogonal projector onto $\mathcal{R}$

$$\longrightarrow \|\bar{v}^h\| = \|K^h v^h\|$$
$$= \|v^h - (I^h - K^h) v^h\|$$
$$= \min_{\delta^h \in \mathcal{R}} \|v^h + \delta^h\|$$

MG5-6

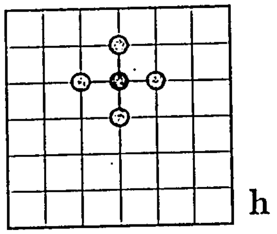
Model problems

The performance of a concrete multigrid method is not sensitive to modifications in the problem, unless its "nature" changes essentially.

Some typical model problems

Poisson equation	$-u_{xx} - u_{yy} = f$
Anisotropic equations	$-\varepsilon u_{xx} - u_{yy} = f$
	$0 < \varepsilon \ll 1$ or $\varepsilon \gg 1$
Discontinuous coefficients	$-(a u_x)_x - (b u_y)_y = f$
Singular perturbations	$-\varepsilon \Delta u + a u_x + b u_y = f$
	$0 < \varepsilon \ll 1$
Indefinite problems	$-\Delta u - c u = f$
Nearly-singular problems	$c > 0$
Eigenvalue problems	
Higher order equations	$\Delta\Delta u = f$

Anisotropic problems (2D)



$$-\varepsilon U_{xx} - U_{yy} = f(x,y)$$

$$L^h = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix}_h$$

Gauss-Seidel relaxation

For all grid points do

$$u^h_{i,j} = \frac{1}{2(1+\varepsilon)}\left[h^2 f^h_{i,j} + \varepsilon u^h_{i-1,j} + \varepsilon u^h_{i+1,j} + u^h_{i,j-1} + u^h_{i,j+1} \right]$$

Smoothing analysis (standard coarsening):

$\mu \to 1$ if $\varepsilon \to 0$ or $\varepsilon \to \infty$

Behavior of error $v^h = U^h - u^h$ during relaxation:

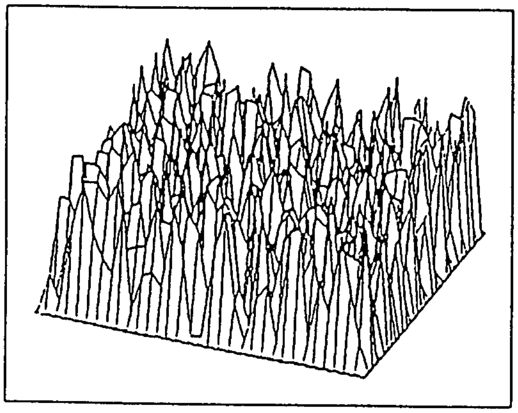
$$v^h_{i,j} = \frac{1}{2(1+\varepsilon)}\left[\varepsilon v^h_{i-1,j} + \varepsilon v^h_{i+1,j} + v^h_{i,j-1} + v^h_{i,j+1} \right]$$

If $0 < \varepsilon << 1$: Smoothing only in y-direction!
If $\varepsilon >> 1$: Smoothing only in x-direction!

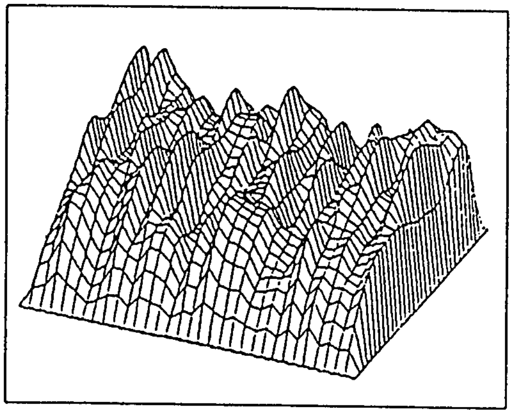
Influence of (pointwise) Gauss-Seidel relaxation on the error

$$-\varepsilon U_{xx} - U_{yy} = f, \quad 0 < \varepsilon \ll 1$$

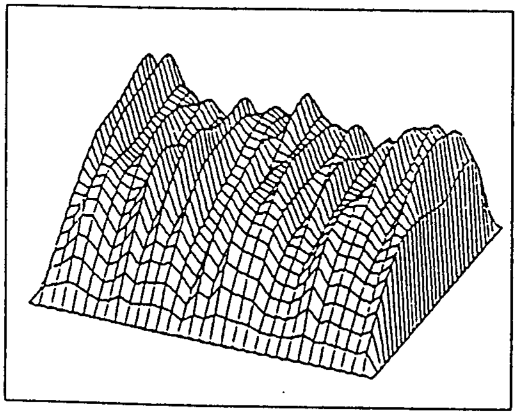
uniform mesh



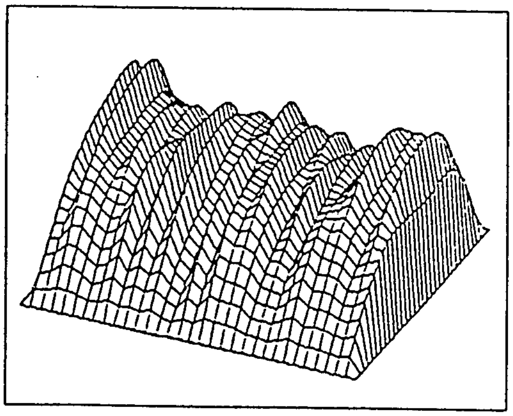
Error of initial guess



Error after 5 relaxations



Error after 10 relaxations



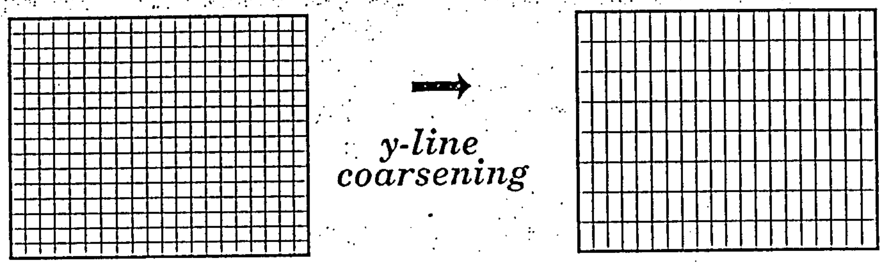
Error after 15 relaxations

~~Aniso1-3~~
M 663-7

Remedy 1: Semi-coarsening

Use Gauss-Seidel (pointwise) relaxation BUT
coarsen only in the direction of error smoothness
(i.e. in the direction of *strong couplings*)

<u>For example:</u> $0 < \varepsilon \ll 1$



$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix} \qquad \frac{1}{4h^2}\begin{bmatrix} & -1 & \\ -4\varepsilon & 2(1+4\varepsilon) & -4\varepsilon \\ & -1 & \end{bmatrix}$$

The anisotropy *gets weaker* on coarser grids!

What to do for *arbitrarily varying* $\varepsilon = \varepsilon(x,y)$?

- **Adaptive coarsening**
  Technically rather complicated (cf. AMG)
- **"Multiple" coarsening**
  Use more than one coarse grid **on the same level,** and combine the corresponding corrections properly.

Remedy 2: Line relaxation

Use standard coarsening BUT

relax all strongly coupled points simultaneously
(*line relaxation* in the direction of *strong couplings*)

If $0 < \varepsilon \ll 1$: Gauss-Seidel y-line relaxation

If $\varepsilon \gg 1$: Gauss-Seidel x-line relaxation

What to do for *arbitrarily varying* $\varepsilon = \varepsilon(x,y)$?

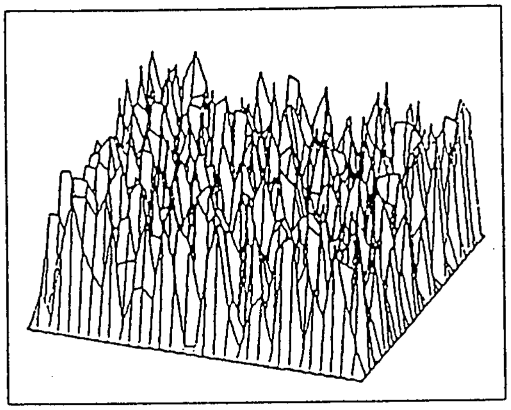
Alternating line relaxation

One step of **x-line** relaxation followed by one
one step of **y-line** relaxation.

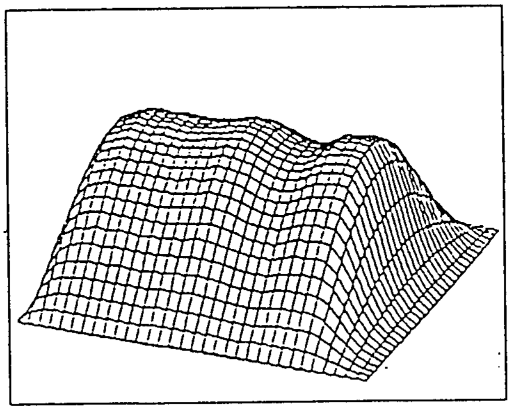
Influence of Gauss-Seidel y-line relaxation on the error

$$-\varepsilon U_{xx} - U_{yy} = f, \quad 0 < \varepsilon \ll 1$$

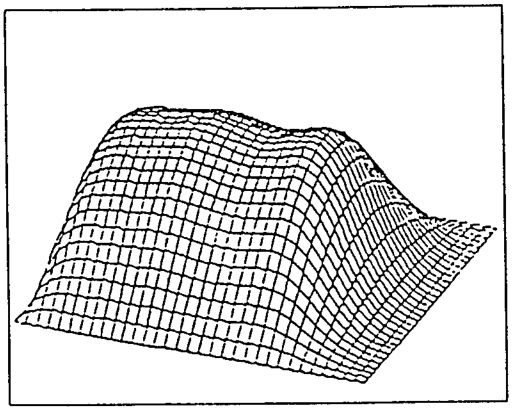
uniform mesh



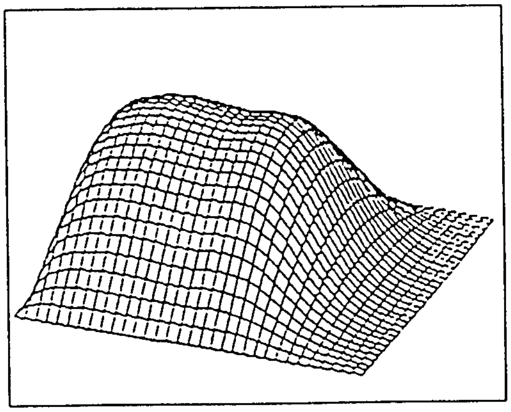
Error of initial guess



Error after 5 relaxations



Error after 10 relaxations



Error after 15 relaxations

~~Anisol-6~~
MG63-6

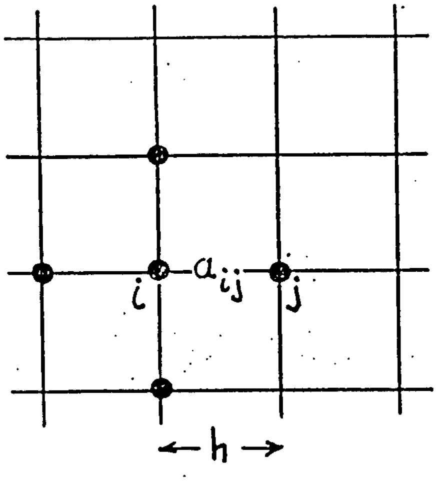
A. Brandt

Multigrid Monte-Carlo and Stochastic Coarsening

OBJECTIVES	MS
Energy minimization	
• Near-minimum start	MG, AMG
• Far start	ms annealing
• Homogenization	
Equilibrium statistics	
• O(n) per sample	ms
• O(1) per sample	Monte-Carlo
⇓	
• Homogenization	
Dynamics	
• Large Δt	MG, AMG
• Very large Δt	ms annealing
Stochastic dynamics	⇐

ms Monte-Carlo at each time step

$$E(u) = \frac{1}{2}\sum_{<i,j>} a_{ij}\left(\frac{u_i - u_j}{h}\right)^2 - \sum_i f_i u_i$$



$$E(U) = \min_u E(u)$$

$$0 = \left.\frac{\partial E}{\partial u_i}\right|_{u=U} = \frac{1}{h^2}\sum_{<j,i>} a_{ij}(u_i - u_j) - f_i$$

$$\frac{1}{h^2}\sum_{<j,i>} a_{ij}(U_i - U_j) = f_i$$

$$\longrightarrow \frac{\partial}{\partial x}\left(a\frac{\partial U}{\partial x}\right) + \frac{\partial}{\partial y}\left(a\frac{\partial U}{\partial y}\right) = f$$

$$\Longrightarrow E(U) = \min E(u)$$

$$E(u) = \iint \left[\frac{a}{2}(u_x^2 + u_y^2) - fu\right] dx\,dy$$

Physical distribution

$$P_E(u) = \frac{1}{Z} e^{-\beta E(u)}, \qquad \beta = \frac{1}{k_B T}$$

$$Z = \sum_u e^{-\beta E(u)}$$

Observables

$$E(u)$$

$$M(u) = \sum_i u_i \,, \quad M(u)^2$$

Averages

$$<M> = \sum_u P_E(u)\, M(u)$$

$$<M^2>, <E>$$

How to compute ?

Monte-Carlo (MC)
Markov chains

$$u^{(0)}, u^{(1)}, \dots, u^{(n)}, \dots$$

$$P_n(u) = P(u^{(n)} = u)$$

$$= \sum_{u'} P_{n-1}(u') \underline{P_{n-1}^{n}(u' \to u)}$$

$P_0(u)$ given (e.g., random)

Detailed balance (DB)

$$\frac{P_{n-1}^{n}(u' \to u)}{P_{n-1}^{n}(u \to u')} = \frac{P_E(u)}{P_E(u')} \quad \text{or} \quad \frac{0}{0}$$

$$\Longrightarrow \lim_{n \to \infty} P_n(u) = P_E(u) \qquad \forall u$$

$$\langle M \rangle = \lim_{k \to \infty} \frac{1}{k} \sum_{n=1}^{k} M(u^{(n)})$$

Metropolis

current u $\longleftrightarrow$ one candidate u'

$$P(u \to u') = \min\left(1, e^{\beta[E(u)-E(u')]}\right)$$

$\Longrightarrow$ Detailed balance

Point-by-point Metropolis

extremely slow to produce a new independent sample

A. Simple Energy Basins

real u_i

Slow to equilibrate and be sampled : $I_H^h V^H$

Coarse Monte-Carlo Hamiltonian: $E^h(\tilde{u}^h + I_H^h V^H)$

$$\equiv E^H(V^H)$$

Multigrid cycle :

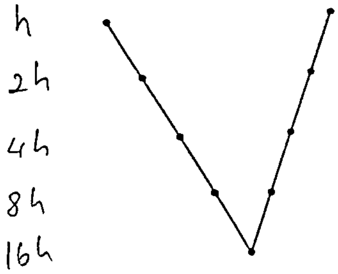
on each level :

1. Monte-Carlo passes
2. γ cycles on coarser
3. Monte-Carlo passes

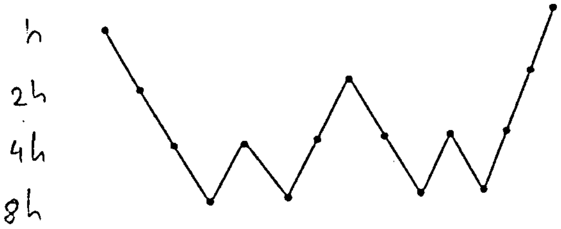
Near equilibration and decorrelation in one cycle ($\gamma = 2$) .

Small work on coarser grids $(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots)$

$\gamma = 1$



$\gamma = 2$



Independent sampling on coarser levels

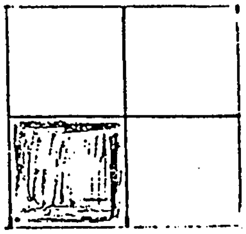
$\Downarrow$

1. **Large-scale statistics:**

   accumulated on coarser levels in $\gamma = 2^d$ cycle

2. **Vast domains:**

   only on coarser levels by domain replication (instead of $\gamma = 2^d$)



3. **Macroscopic dynamics:**

   cheaply, directly

Hamiltonian Simplification

$$H(u) = H'(u) + \Delta(u), \qquad \tilde{u} \xrightarrow[H']{} u$$

1. Apriori lottery

Delete Δ in probability $P_\Delta(\tilde{u}) \sim e^{-\Delta(\tilde{u})/T}$

Freeze Δ in probability $1 - P_\Delta(\tilde{u})$

Local.

Allows coarse dynamics.

Freezing should correspond to I_H^h

2. Aposteriori lottery

$$P_{accept}(\tilde{u} \to u') = \begin{cases} 1 & \text{if } \Delta(\tilde{u}) > \Delta(u') \\ e^{\Delta(\tilde{u}) - \Delta(u')} & \end{cases}$$

Global, convenient

Requires small $\Delta(u)$ for smooth $u - \tilde{u}$

Prohibits coarse dynamics?

B. Ising Spins

$u_i = \pm 1$

$$E(u) = -\sum_{<i,j>} J_{ij} u_i u_j \qquad P(u) \sim e^{-E(u)/T}$$

- Coarse moves = block flips.
- Standard blocks are improbable.
- Blocks should have uniform sign.
- Basing blocks on current u disrupts statistical fidelity

$$\frac{P(u_1 \to u_2)}{P(u_2 \to u_1)} = \frac{P(u_2)}{P(u_1)}$$

(detailed balance).

Stochastic coarsening at $\tilde{u}$

- Delete $V_{ij}(u) = -J_{ij} u_i u_j$ from E in probability $p_{ij} = q_{ij} e^{-V_{ij}(\tilde{u})/T}$

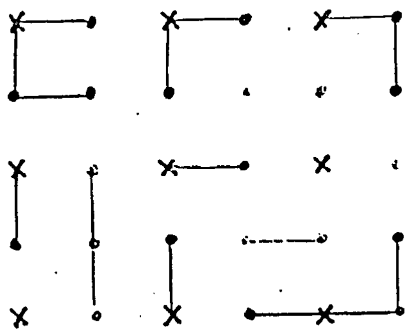
Freeze V_{ij} (block i with j) if undeleted.

⇒ detailed balance. [Swendsen-Wang]

Process all interactions.

Also between blocks.

Except between any two "coarse spins".



- Coarse Hamiltonian $E^1 = -\sum J^1_{ij} v_i v_j$
  + "independent blocks".

Repeat: Hamiltonians $E^0, E^1, E^2, E^3, \dots$

On meshsizes $1, 2, 4, 8, \dots$

+ accumulating list of independents.

μ-cycle (V, W, ...)

- Multi-scale island dynamics.

Fast sampling. But no independence of coarse samplings.

Detailed Balance

$$E(u) = -\mathbf{V}(u) + E'(u)$$

Probability of deleting $\mathbf{V}$ given $\tilde{u}$:

$$p_V(\tilde{u}) = q_V e^{-V(\tilde{u})/T} \qquad = q_V \frac{Z' \, P_{E'}(\tilde{u})}{Z \, P_E(\tilde{u})}$$

<u>otherwise:</u> freeze V

If $\mathbf{V}(\tilde{u}) \neq \mathbf{V}(u)$:

$$\frac{P(\tilde{u} \to u)}{P(u \to \tilde{u})} = \frac{p_V(\tilde{u}) \, P_{E'}(\tilde{u} \to u)}{p_V(u) \, P_{E'}(u \to \tilde{u})} = \frac{p_V(\tilde{u})}{p_V(u)} \, \frac{P_{E'}(u)}{P_{E'}(\tilde{u})}$$

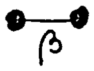
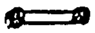
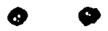
$$= \frac{P_E(u)}{P_E(\tilde{u})}$$

If $\mathbf{V}(\tilde{u}) = \mathbf{V}(u)$: $\qquad p_V(\tilde{u}) = p_V(u) = p$

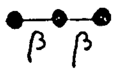
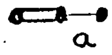
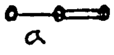
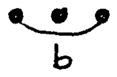
$$\frac{P(\tilde{u} \to u)}{P(u \to \tilde{u})} = \frac{p \, P_{E'}(\tilde{u} \to u) + (1-p) \, \hat{P}_E(\tilde{u} \to u)}{p \, P_{E'}(u \to \tilde{u}) + (1-p) \, \hat{P}_E(u \to \tilde{u})}$$

$$= \frac{P_{E'}(u)}{P_{E'}(\tilde{u})} = \frac{P_E(u)}{P_E(\tilde{u})}$$

2-spin coarsening (SW)

2-spin configuration	given H	alternative $\mathcal{H}_1$	H_2	transition pr. P_1	P_2
$+ \quad +$	$-\beta$	∞	0	$1-e^{-2\beta}$	$e^{-2\beta}$
$+ \quad -$	β	0	0	0	1
					

3-spin coarsening (example)

3-spin configur.	given H	alternative H_1	H_2	H_3	transition probab. P_1	P_2	P_3
$+ \quad + \quad +$	-2β	$-a$	$-a$	$-b$	$\frac{1-e^{-4\beta}}{2}$	$\frac{1-e^{-4\beta}}{2}$	$e^{-4\beta}$
$+ \quad - \quad +$	2β	∞	∞	$-b$	0	0	1
$+ \quad + \quad -$	0	a	∞	b	$1-P_*$	0	P_*
$+ \quad - \quad -$	0	∞	a	b	0	$1-P_*$	P_*
							

)etail balance $\Leftarrow$ $e^{2a} = \dfrac{e^{2\beta} - e^{-2\beta}}{2(1-P_*)}$, $\quad e^{2b} = \dfrac{e^{-2\beta}}{P_*}$

GRID	$\langle M^2 \rangle$		
	fine level	coarse-level deviation	
		2-spin (SW) coarsening	3-spin (p_* = .15) coarsening
4 × 4	12.2	± 1.8	± .7
8 × 8	41.4	± 7.2	± 1.5
16 × 16	139.5	± 25.6	± 4.0
32 × 32	470.2	± 81.6	± 10.6

25% sites treated

Stochastic Dynamics

At time t^n : position vector X^n
velocity vector V^n
potential energy $E(X^n)$.

Deterministic implicit time step :

$$-\nabla E(X^{n+1}) = p^{n+1}$$

$$p_k^{n+1} = m_k \frac{\delta V_k}{\delta t} , \ \delta V_k = V_k^{n+1} - V_k^n , \quad V^{n+1} = \frac{X^{n+1} - X^n}{\delta t}$$

$\Longleftrightarrow$ X^{n+1} minimizes $H_n(X^{n+1})$

$$H_n(X^{n+1}) = \frac{1}{2} \Sigma \, m_k (\delta V_k)^2 + E(X^{n+1})$$

Stochastic time step :

$$P(X^{n+1}) \sim e^{-\beta H_n (X^{n+1})}$$

$$\beta = \frac{1}{k_B T}$$

Results of the stochastic step :

- $\langle p^{n+1} \rangle = -\nabla E(x^{n+1})$
- $\langle (p_k^{n+1})^2 \rangle - \langle p_k^n \rangle^2 = m_k / \beta$
- Local components, with oscillation period $\ll \delta t$ are fully thermalized:

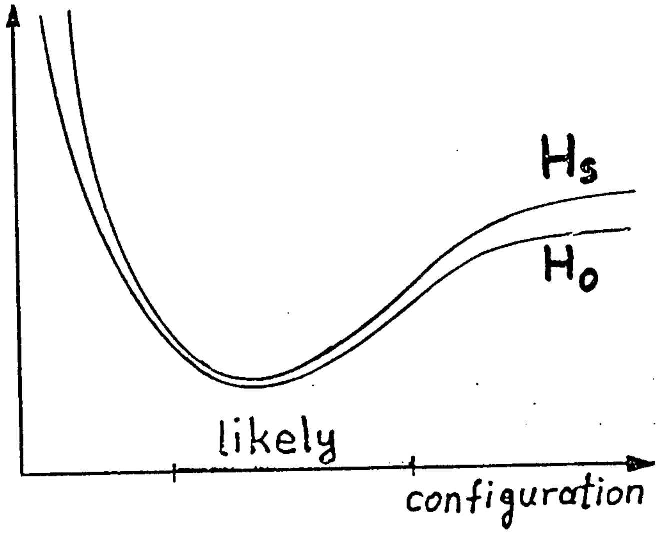
$$P(x^{n+1}) \sim e^{-E(x^{n+1})/k_B T}$$

- Large scale components, with oscillation period $\gg \delta t$ satisfy Newton law.

- Easier multiscaling

Stochastic Coarse-Hamiltonian Simplification

STOCHASTIC HAMILTONIAN SIMPLIFICATION



$\mathcal{H}_0$ = Original Hamiltonian
obtained from fine grid ⟹ complicated

$\mathcal{H}_S$ = Simplified Hamiltonian
FAS: original form + polinomial terms

$$\mathcal{H}_s \geqslant \mathcal{H}_0 \,,\quad \mathcal{H}_s \approx \mathcal{H}_0$$

Detailed Balance: $P(H_o \rightarrow H_S) = e^{H_o - H_S}\Big|_{\text{current}}$

$P(\text{freeze } H_S - H_o) = 1 - P(H_0 \rightarrow H_s)$

Simpler : Approximate detailed balance

Thermodynamic limit

Statistical average of an infinite system: $N \to \infty$

Calculation to accuracy $\varepsilon\sigma$

$\Rightarrow$ Large $N = n^d$ points

$\times$ n^2 Monte-Carlo sweeps to produce a sample

$\times$ $O(\varepsilon^{-2})$ samples

$[\times$ calculation of $\delta \log \det A]$

$= O(n^{2+d}\varepsilon^{-2})\ [\times \cdots]$

MG calculation in $O(\varepsilon^{-2})$

Statistical Fields

$$\text{Probability}(u) \sim e^{-\text{Energy}(u)/T}$$

Monte Carlo: particle-by-particle (or point-by-point) simulation

Slow large-scale sampling:

1. Small changes per sweep
2. Few samples per configuration

Multigrid Monte Carlo:

1. Moves on all scales
2. Much sampling on coarse levels

E.g., Gaussian model on $L \times \cdots \times L = L^d$ lattice:

$O(1)$ instead of $O(L^{d+2})$ operations per sample

$\Longrightarrow$ Homogenization

A. Brandt

Global and Discrete-State Optimization: Multiscale Annealing

min $E(u)$

False attraction basins

Particle by particle minimization
trapped in local attraction basins.

Simulated annealing $\quad P(u) \underset{T \to 0}{\sim} e^{-E(u)/T}$

trapped in large-scale attraction basins.

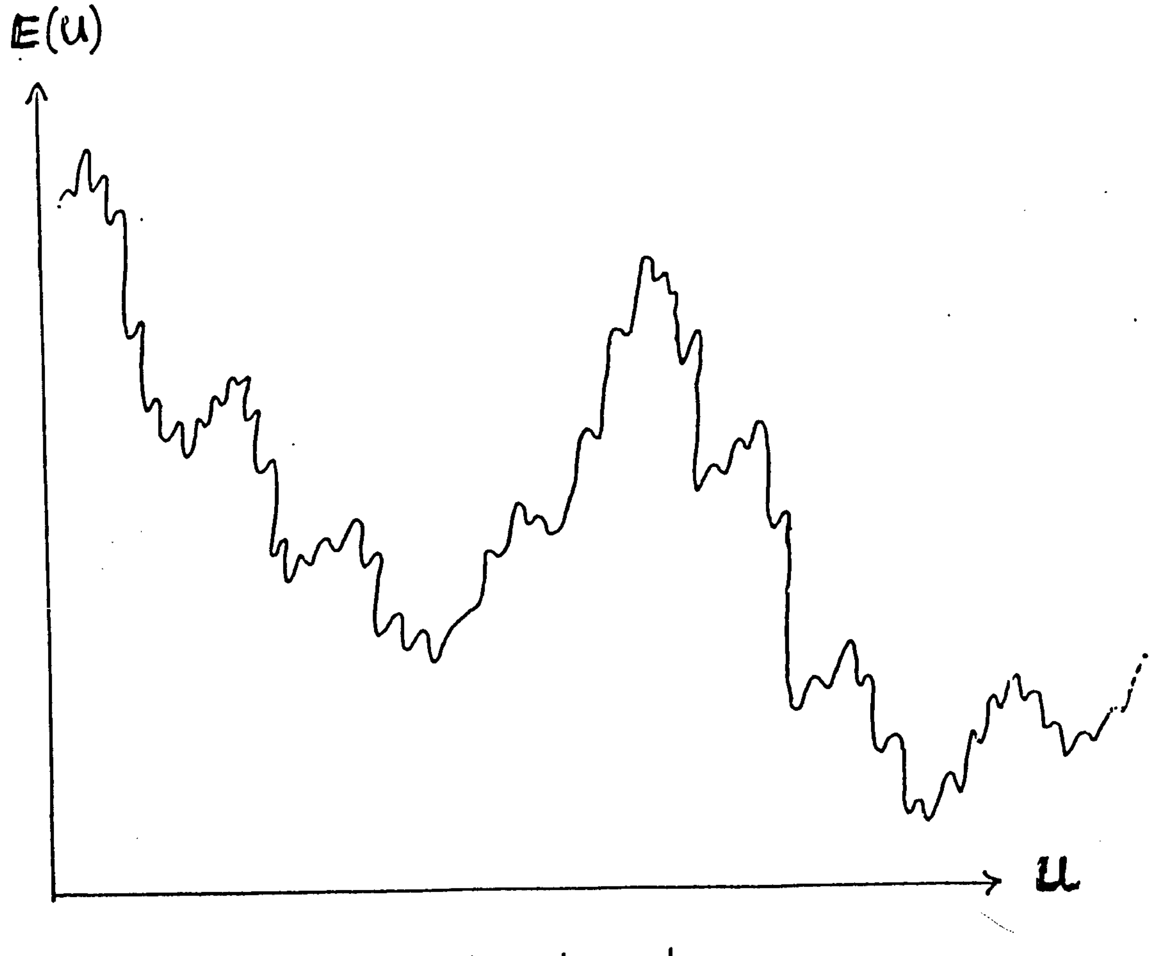
Multi-level annealing
Collective moves, at all scales
converges (fast, in probability).

- A large-scale move is decided only after optimizing around it at all finer scales.
- Fast annealing at each level.
- Recombinations at each level.

Optimization   min $E(u)$

- Fast convergence near optimum

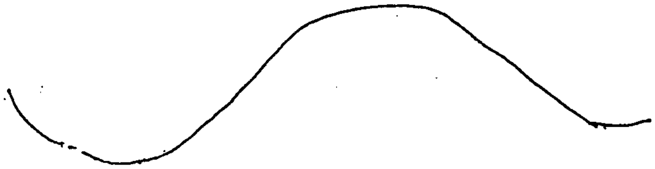
- Global optimization
  Escape false attraction basins

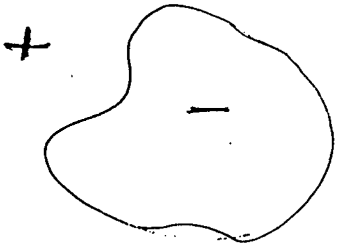


multi-scale attraction basins

Multi-level stochastic optimization (minimization)

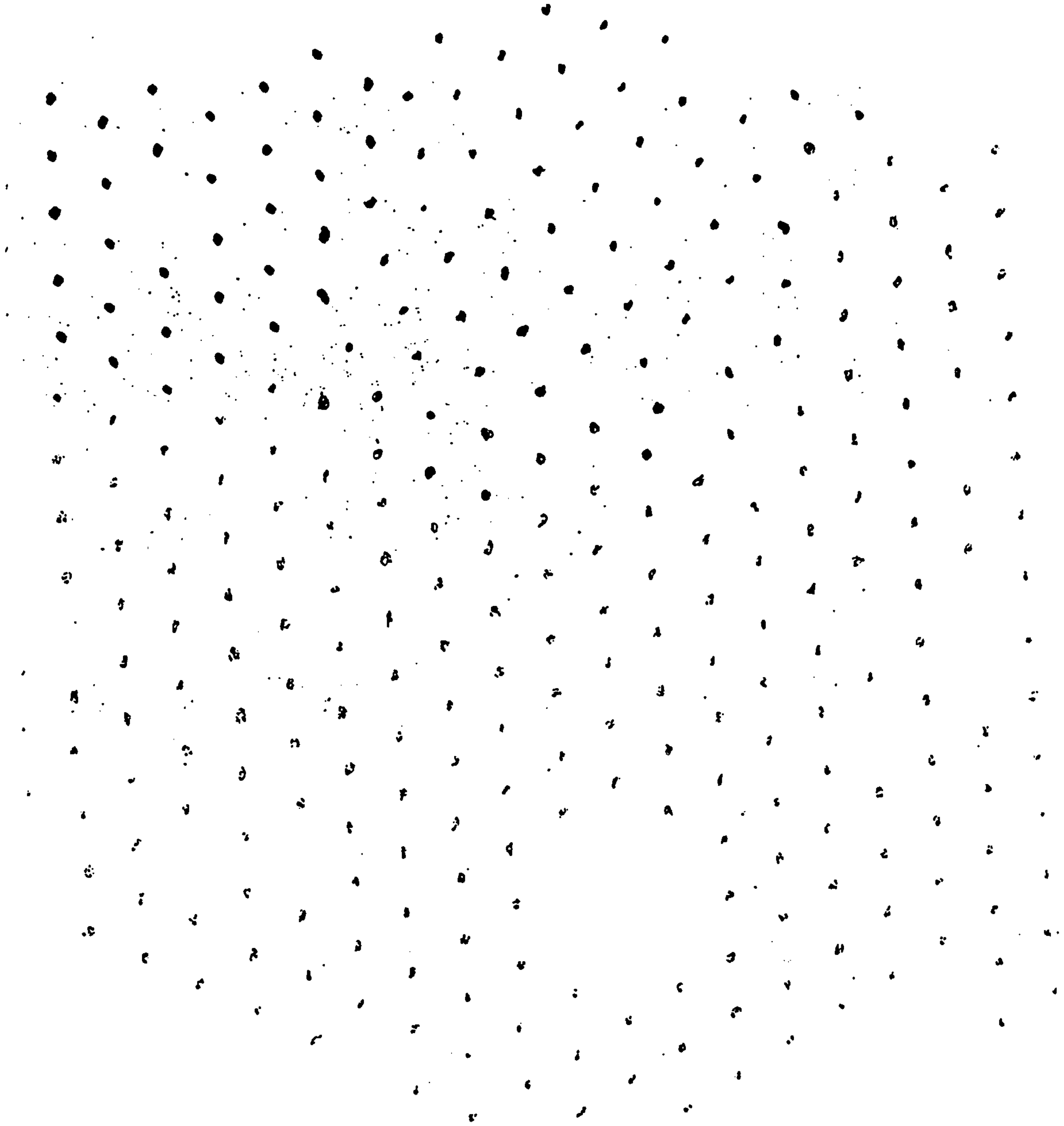
Large basins ⇔ large-scale spatial features





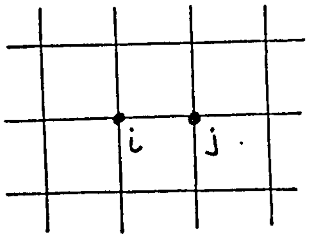
Large scale changes decided (stochastically) only after optimizing around them at all finer scales

Recursion



$$P(u) = \frac{1}{Z} e^{-E(u)/T}$$

$$E(u) = \sum_{\langle i,j \rangle} (u_i - u_j)^2 - \sum_i f_i u_i$$

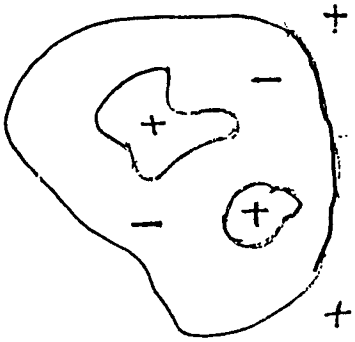


Real u_i: $f_i = 0 \Rightarrow u_i \approx u_j$

O_1 : Ising spins $u_i = \pm 1$

Islands within islands

Hierarchy of basins



O_2, O_3, … spins : vectors $|u_i| = 1$

Many Local Minima*)

Ising spins: $\min\limits_{s_i = \pm 1,\ \text{BC}} E(s)$

spin glasses

$$E(s) = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j - \sum_i h_i s_i$$

Relaxation point-by-point minimization:

Flipping a spin if the energy decreases

Artificial temperature $\frac{1}{\beta}$ $P(s) \sim e^{-\beta E(s)}$

Point-by-point stochastic relaxation

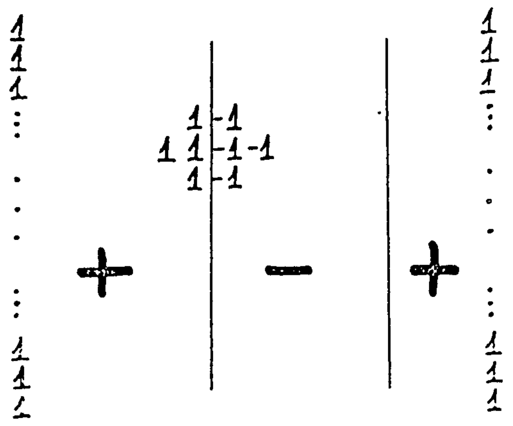
Annealing: Increase β gradually

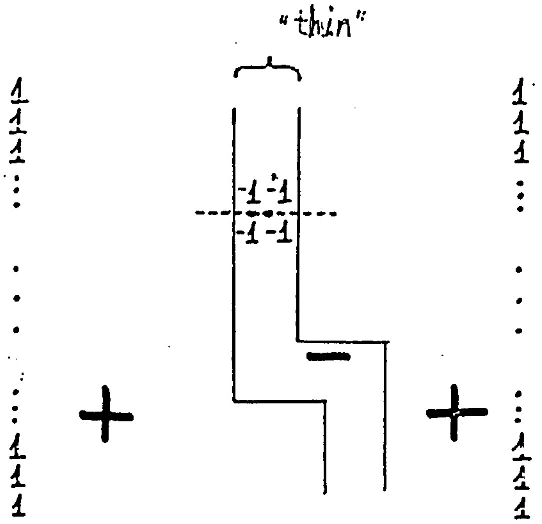
*) with Dorit Ron (1985)

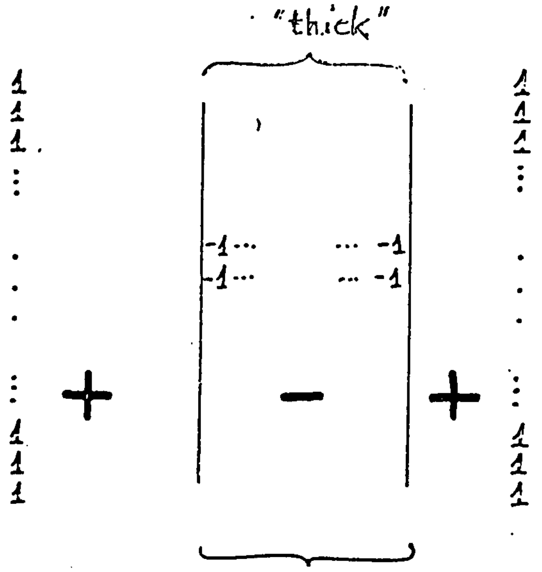
Ising Model

$$E(S) = -\sum_{\langle i,j \rangle} S_i S_j$$

- Ground-State : $S_i \equiv 1$

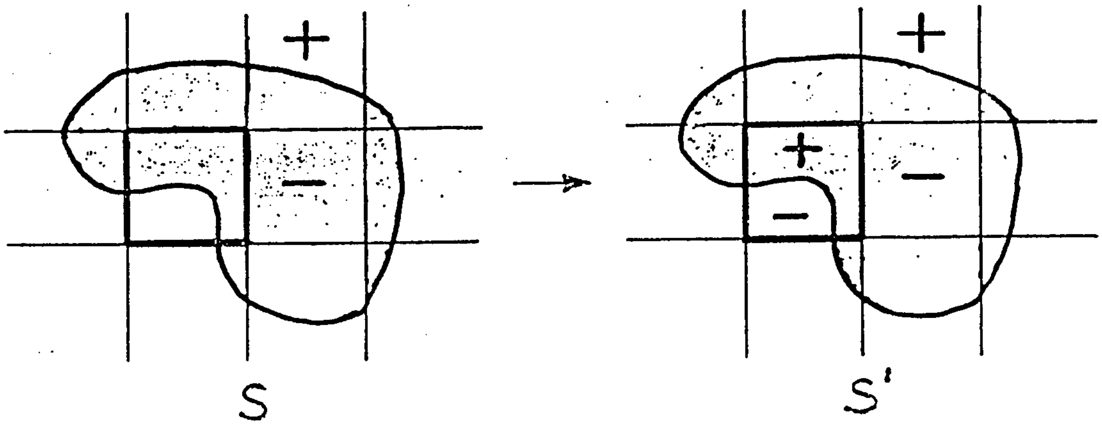






- External Magnetic Feild : $H < 0$
- Ground-State remains : $S_i \equiv 1$

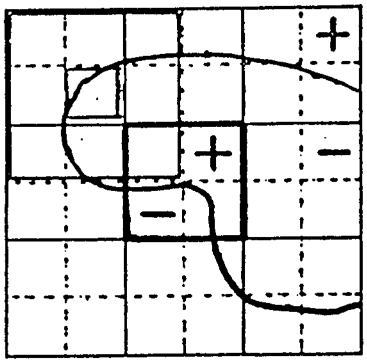
Block-by-block Relaxation



$$E(S) < E(S')$$

$\Longrightarrow$ Local optimization around S'

Working Area:



Block revision process $\Longrightarrow$ revised block

PRINCIPLES of DISCRETE-STATE and highly non-quadratic OPTIMIZATION

1. Hierarchy of change scales

2. Large-scale change is decided only after calculating its effects (optimizing around it) at all finer scales, starting from the finest.

3. At each scale employ stochasticity just large enough to escape local minima on that scale, then strict minimization.

4. Repeat (the finer - the more) with LCC

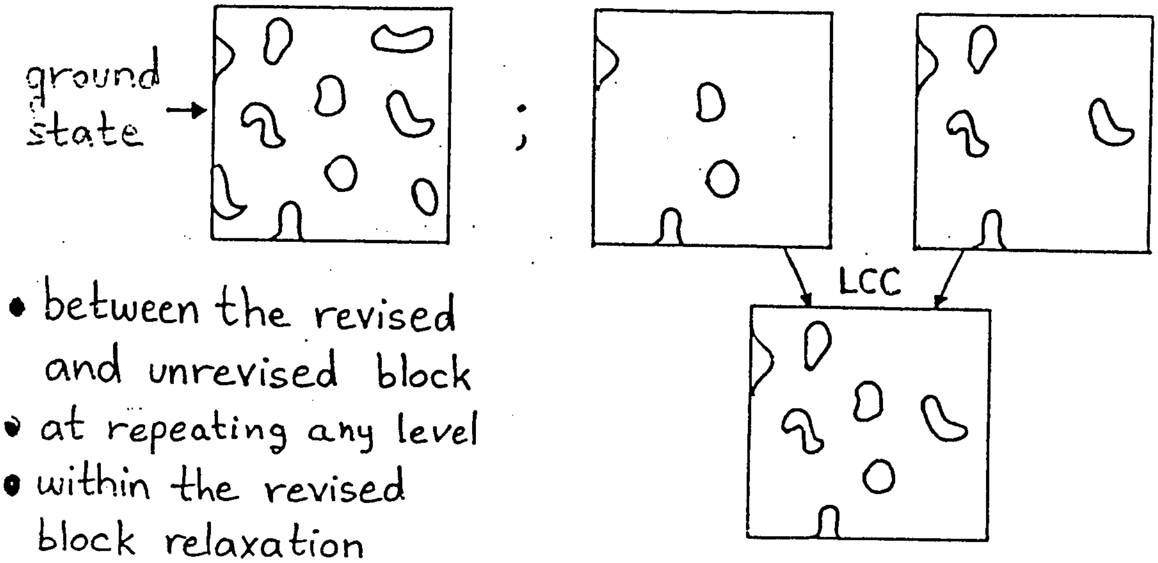
5. Recursion

- Acceleration ○ Hyper parallel

- Escaping local minima with large attraction basins

Supplementary Minimization Techniques

1. Lowest Common Configuration - LCC



- between the revised and unrevised block
- at repeating any level
- within the revised block relaxation

2. Lower starting β

- increases the probability of breaking long blocks

3. Adaptive relaxation

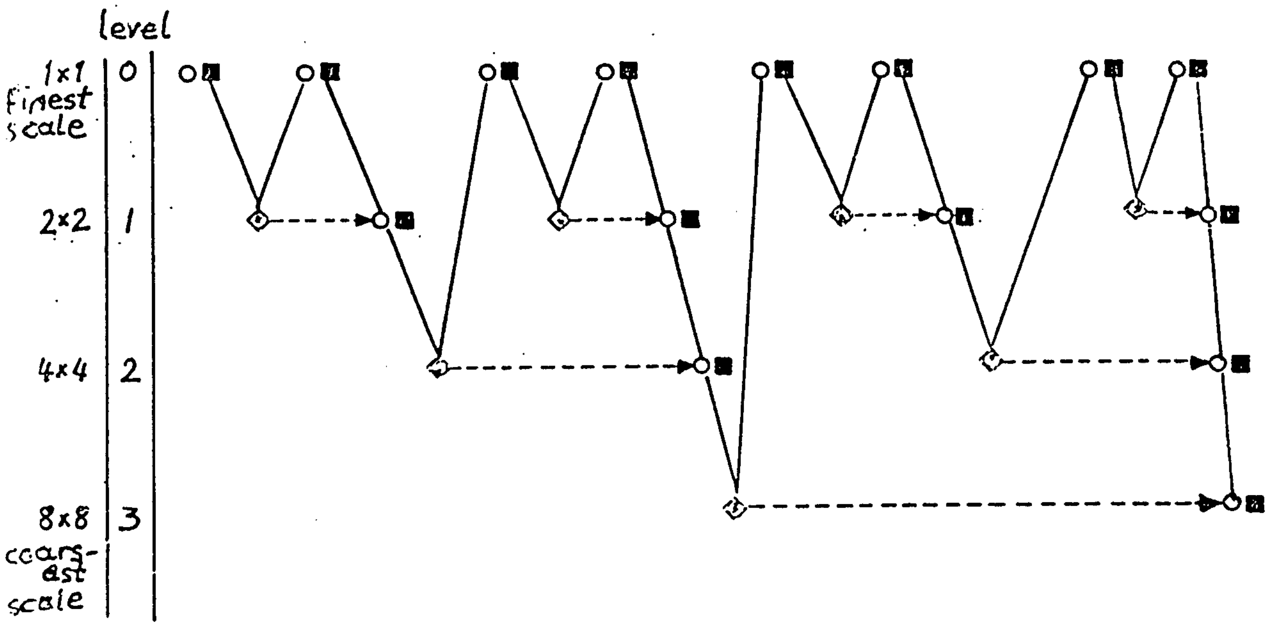
- saves work
- relaxation may expand beyond original work areas

4. Position shifts

- half-size of the blocks

5. New type of blocks : broken along weak bonds

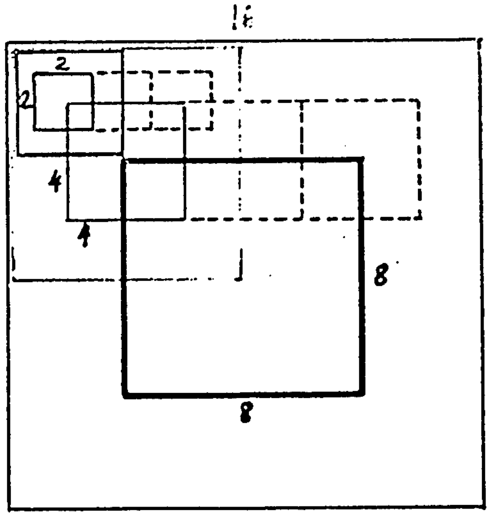
EXAMPLE: A complete multilevel cycle



◈ flipping a square of spins

○ stoshastic relaxation

□ minimization relaxation



Complexity: $O(N^{3/2})$; never exceeds $O(N^2)$

Linear Programming

Transportation Problem

Lumping 2 neighboring
destinations, origins, ...
blocks, super-blocks

Coarse costs
= fine marginals

FMG-like solver

Total work
≈ 1 simplex (MODI) step

Role of relaxation [Kaminsky]

MULTI-LEVEL OPTIMIZATION TECHNIQUES

Equations + Optimization

1-FMG : Parameter optimization on coarse
Local optimization at relaxation

Interactive design: Re-solving mostly on coarse

optimal
control

Functional extremization

Quadratic functional (⇒ linear equations)

Constrained quadratic: FAS $u_{ij} \geq 0$
$|u_{ij}| \leq 1$

Non quadratic (few extrema): FAS

Highly non-quadratic
Discrete states $u_{ij} = \pm 1$ } (many extrema): multi level annealing

Linear programming transportation problem

Kaminsky

Full multigrid (FMG) and nonlinear multigrid

Klaus Stüben

GMD/SCAI
Schloß Birlinghoven
D-53757 St. Augustin
Germany

Goal of FMG:
O(N) logN → O(N)

Full multigrid (FMG)

Problems with iterative methods:

First guess? Stopping criterion?

First guess?

Use coarse-grid approximations as first guesses for iterations on finer grids.

→ ***nested iteration***

Stopping criterion?

$$\rightarrow \underbrace{\|U - u^h\|}_{\textit{approximation error}\ \text{(of current approximation)}} \approx \underbrace{\|U - U^h\|}_{\textit{discretization error}} \quad (*)$$

Generally, it makes no sense to iterate the ***algebraic error*** such that

$$\underbrace{\|U^h - u^h\|}_{\textit{algebraic error}} \ll \underbrace{\|U - U^h\|}_{\textit{discretization error}}$$

FMG is, formally, just a combination of

MG cycling + nested iteration

However, ...

Since cycle convergence is *h-independent*:

(*) is naturally obtainable also;

Total FMG work: O(N)

Full multigrid algorithm

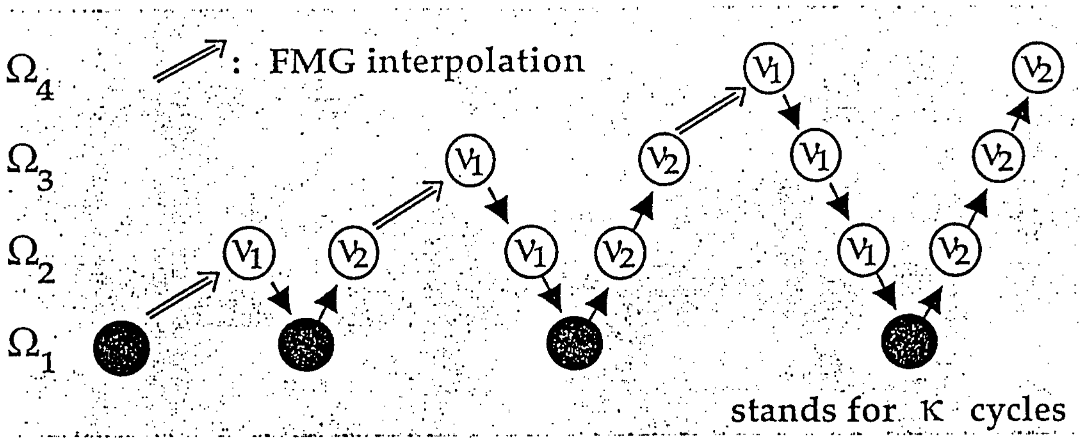
Sequence of m increasingly finer grids

$$\Omega_1, \Omega_2, \ldots, \Omega_m \; (=\Omega_h)$$

```
For k=1,2,...,m do
    If k=1 then
        Solve $L^k U^k = f^k$
    else
        $u^k = \Pi^k_{k-1} u^{k-1}$
        $u^k \leftarrow MG^{\kappa}(u^k; L^k, f^k)$
    endif
enddo
```

Π^k_{k-1}: denotes higher order interpolation *(FMG interpolation)*.

$\kappa>0$: denotes the number of MG-cycles to be performed.



Order of the FMG interpolation:
at least p (= order of discretization)

Standard coarsening, 2nd order discretizations:
cubic interpolation

FMG performance

ρ := MG-cycle convergence factor

$E_k := \|U - U^k\|$ discretization error $\qquad e_k := \|U - u^k\|$ current approximation error

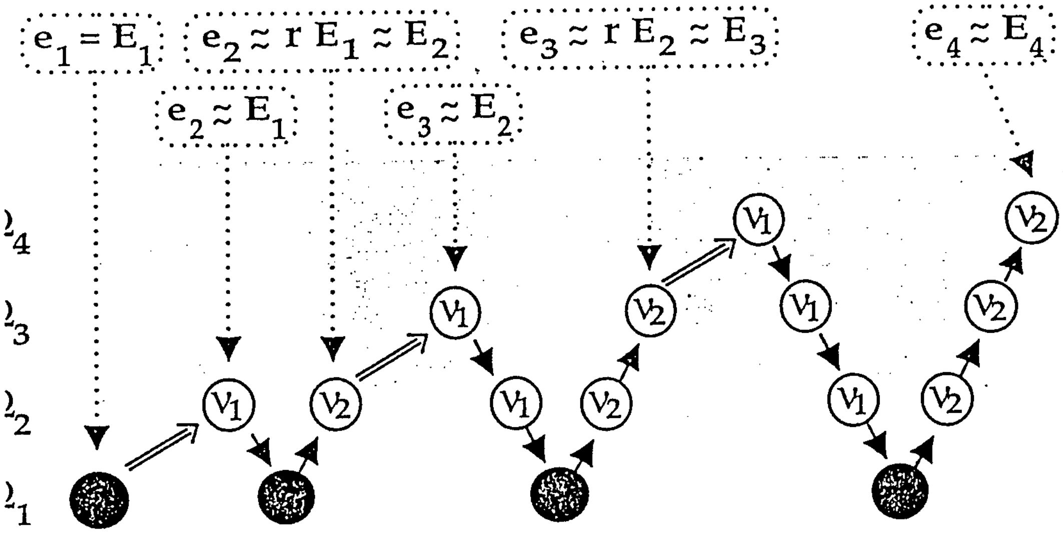
Goal: $e_k \approx E_k$

Assumption: $E_k \approx r E_{k-1}$

Roughly:

For each k, the final approximation u^k is accurate up to the level of the discretization error, i.e. $e_k \approx E_k$, if

$$\rho^{\kappa} < r$$



stands for κ cycles

Work count in 2D, standard coarsening

$\mathcal{W}_{\text{cycle}}$:= work per cycle *on finest level*

$$\mathcal{W}_{\text{FMG}} = \kappa\, \mathcal{W}_{\text{cycle}} \,(1 + 1/4 + 1/16 + \ldots) = 4/3\ \kappa\, \mathcal{W}_{\text{cycle}}$$

ignored: FMG interpolation

FMG is true O(N) method

Poisson equation:

V(2,1)-cycle, $\rho \approx 0.1$, $r = 0.25$ ⟶ $\kappa = 1$

$\mathcal{W}_{\text{FMG}} \approx 4/3\ \mathcal{W}_{\text{cycle}} \approx 7$ relax on finest grid

Solving Poisson-like problems with FMG
costs approximately the equivalent of
7 relaxations on the finest grid!

Remarks

$L^k U^k = f^k$ should "make sense" on the coarser grids

Otherwise, start FMG on a **higher level** or
use **"averaged equations"** on the lower levels.

Nonlinear multigrid (FAS)

Nonlinear $L^h U^h = f^h$, approximate solution: u^h

$L^h V^h = d^h$ makes no sense any more

"Naive" approach

Global linearization (outer iteration) + linear MG (inner iteration)

- Linearization (e.g. Newton) may be complex
- The inner + outer iterations have to be matched

"Direct" approach

- No global linearization!
  (Local) linearization only in relaxation

  ⟶ No matching of different iterations required
  ⟶ Same cycle structure as for linear problems
  ⟶ Sometimes easier to converge
- Multigrid performance as for linear problems
- Easy incorporation of "special techniques"
  (adaptive multigrid, τ -extrapolation, ...)

Nonlinear correction equation

Instead of $L^h V^h = d^h$

$$\boxed{\begin{array}{c} L^h(V^h + u^h) - L^h u^h = d^h \quad \text{with} \quad d^h := f^h - L^h u^h \\ \Longrightarrow \; U^h = u^h + V^h \end{array}}$$

Approximation w.r.t. Ω_H and L^H ?

$$L^H(V^H + \check{I}_h^H u^h) - L^H(\check{I}_h^H u^h) = I_h^H d^h$$

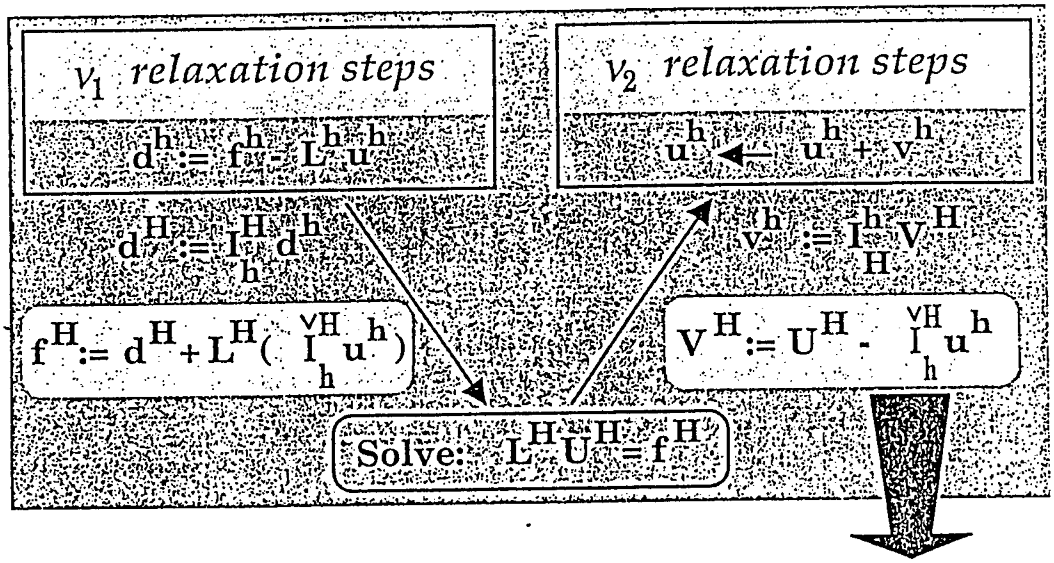
$$\Longrightarrow \; L^H\underbrace{(V^H + \check{I}_h^H u^h)}_{=:\, U^H} = \underbrace{I_h^H d^h + L^H(\check{I}_h^H u^h)}_{=:\, f^H}$$

$$\Longrightarrow \; L^H U^H = f^H$$

$$\boxed{V^H := U^H - \check{I}_h^H u^h}$$

- $\check{I}_h^H$ is a transfer operator which may differ from I_h^H
- Note that U^H and f^H are *defined* above. They are NOT the solutions and right hand side, resp., of the H-discretized continuous problem!

Nonlinear two-grid cycle



ν_1 *relaxation steps*

$d^h := f^h - L^h u^h$

$d^H := I_h^H d^h$

$f^H := d^H + L^H(\check{I}_h^H u^h)$

Solve: $L^H U^H = f^H$

$V^H := U^H - \check{I}_h^H u^h$

$v^h := I_H^h V^H$

ν_2 *relaxation steps*

$u^h \longleftarrow u^h + v^h$

Only *smooth quantities* (i.e. errors) must be interpolated back to the fine grid!! Do not interpolate U^H itself!

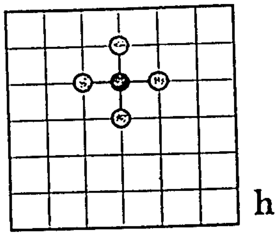
- Smoothing requires a *nonlinear relaxation scheme*
- Extension to "real MG" as in the linear case
- This nonlinear cycle *coincides with the linear one* if the problem is linear
- At convergence:

$$u^h = U^h \quad \text{and} \quad U^H = \check{I}_h^H U^h$$

Standard choice: $\check{I}_h^{2h}$ = *straight injection*

At convergence, all coarse-grid solutions are identical to the finest-grid solution

Nonlinear Example



$$-\Delta U + c(x,y,U) = 0$$

Assumption: $c_u(x,y,U) \geq 0$

Nonlinear Gauss-Seidel relaxation

For all grid points do: compute $\bar{u}^h_{i,j}$ such that

$$\left[4\,\bar{u}^h_{i,j} - u^h_{i-1,j} - u^h_{i+1,j} - u^h_{i,j-1} - u^h_{i,j+1}\right] + h^2 \boxed{c(x,y,\bar{u}^h_{i,j})} = 0$$

and $u^h_{i,j} \longleftarrow \bar{u}^h_{i,j}$

Picard linearization (only good if $c_u(x,y,U^h_{i,j})$ "small"):

$$c(x,y,\bar{u}^h_{i,j}) \longrightarrow c(x,y,u^h_{i,j})$$

Newton linearization:

$$c(x,y,\bar{u}^h_{i,j}) \longrightarrow c(x,y,u^h_{i,j}) + c_u(x,y,u^h_{i,j})\,(\bar{u}^h_{i,j} - u^h_{i,j})$$

→ **Same efficiency as Poisson-solver!**

Further remarks

$$\boxed{\begin{array}{l} \textit{h-truncation error: } \tau_h := L^h U - LU \\ \textit{H-truncation error: } \tau_H := L^H U - LU \end{array}}$$

Coarse-grid equation

$$L^H U^H = \; f^H \; = I_h^H f^h + \underbrace{\boxed{L^H(\check{I}_h^H u^h) - I_h^H L^h u^h}}_{}$$

$$=: \tau_h^H = \textit{(h,H)-relative truncation error}$$

Under certain assumptions

$$\boxed{\tau_h^H \approx \tau_H - \tau_h \;\; = L^H U - L^h U}$$

Applications

τ -estimation

For example: $\tau_H \approx c\,\tau_h \;\Longrightarrow\; \tau_h \approx \tau_h^H / (c-1)$

Stopping criterion: $\|L^h u^h - f\| < \|\tau_h\|$

τ -extrapolation

$$\tau_h^H \longleftarrow \frac{c}{c-1}\,\tau_h^H \;\approx\; \tau_H$$

Adaptive multigrid (MLAT)

Molecular Dynamics at Large Time–Steps

Eric Barth
Margaret Mandziuk
Tamar Schlick
Guihua Zhang

Courant Institute of Mathematical Sciences

Department of Chemistry

New York University

and

Howard Hughes Medical Institute

In molecular dynamics, calculating the N-body forces of interaction dominates computational effort.

$$\begin{aligned} V \;=\; & \sum_{\text{bonds}} k_l \Delta l \\ & + \sum_{\text{angles}} k_\theta \Delta\theta \\ & + \sum_{\text{dihedrals}} k_\phi (1 + cos(n\phi + \delta)) \\ & + \sum_{\text{atom pairs } ij} \left(\frac{q_i q_j}{R_{ij}} - \frac{\alpha_{ij}}{R_{ij}^6} + \frac{\beta_{ij}}{R_{ij}^{12}} \right) \end{aligned}$$

- Standard explicit integrators require one force evaluation per time–step.
- Fastest motions necessitate small steps.

Unfeasible number of force evaluations needed to simulate slow motions over time intervals of interest.

Improved efficiency has been sought by:

- Constraining fast interactions

SHAKE

reduced variable treatments—Rice & Brunger, Moldyn, Oren Becker

- Dividing the forces by time scales

multiple time–step methods—Humphries, Byrne, Watanabe, Karplus

- Super–stable implicit integration schemes

LI—Schlick

- Reference systems for the fast motions.

LIN and related methods—Zhang & Schlick

$$\begin{aligned} V = & \sum_{\text{bonds}} k_l \Delta l \\ & + \sum_{\text{angles}} k_\theta \Delta\theta \\ & + \sum_{\text{dihedrals}} k_\phi (1 + cos(n\phi + \delta)) \\ & + \sum_{\text{atom pairs } ij} \left(\frac{q_i q_j}{R_{ij}} - \frac{\alpha_{ij}}{R_{ij}^6} + \frac{\beta_{ij}}{R_{ij}^{12}} \right) \end{aligned}$$

The Equations of Motion

The Hamiltonian system

$$\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= -\nabla V(q) \end{aligned}$$

Langevin Dynamics

$$\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= -\nabla V(q) - \gamma p + r \end{aligned}$$

Overdamped Langevin Dynamics

(Brownian Dynamics)

$$\begin{aligned} \dot{q} &= M^{-1}p \\ 0 &= -\nabla V(q) - \gamma p + r \end{aligned}$$

Traditional Molecular Dynamics

The Hamiltonian system

$$\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= -\nabla V(q) \end{aligned}$$

Discretization: Velocity Verlet Method

$$\begin{aligned} p_{n+1/2} &= p_n - \frac{h}{2}\nabla V(q_n) \\ q_{n+1} &= q_n + hM^{-1}p_{n+1/2} \\ p_{n+1} &= p_{n+1/2} - \frac{h}{2}\nabla V(q_{n+1}) \end{aligned}$$

Verlet is:

- Explicit
- Second order in time
- Symplectic
- Time reversible
- The gold standard in MD

Constrained Dynamics

The Hamiltonian system with constraints.

$$
\begin{aligned}
\dot{q} &= M^{-1}p \\
\dot{p} &= -\nabla V(q) + g'(q)^t\lambda \\
0 &= g(q)
\end{aligned}
$$

Discretization: Verlet with SHAKE

$$
\begin{aligned}
p_{n+1/2} &= p_n - \frac{h}{2}\nabla V(q_n) + \frac{h}{2}{g'_n}^t\lambda_n \\
q_{n+1} &= q_n + hM^{-1}p_{n+1/2} \\
0 &= g(q_{n+1}) \\
p_{n+1} &= p_{n+1/2} - \frac{h}{2}\nabla V(q_{n+1}) + \frac{h}{2}{g'_{n+1}}^t\lambda_{n+1}
\end{aligned}
$$

A typical bond constraint between atoms i and j:

$$g_\alpha(q) = \frac{1}{2}\left(r_{ij}^2 - L_\alpha^2\right) = 0.$$

Torsion dynamics, multibody treatments, and other variable reduction techniques use generalized coordinates which rules out Verlet integration

Multiple Time–Step Dynamics

The split Hamiltonian system.

$$\begin{aligned}\dot{q} &= M^{-1}p\\ \dot{p} &= -\nabla V_{\mathsf{fast}}(q) - \nabla V_{\mathsf{slow}}(q)\end{aligned}$$

Discretization: reversible-RESPA

```
$q_0 = q_n$
$p_0 = p_n - \frac{kh}{2}\nabla V_{\mathsf{slow}}(q_n)$
for $i = 0 : k-1$,
    $p_{i+1/2} = p_i - \frac{h}{2}\nabla V_{\mathsf{fast}}(q_i)$
    $q_{i+1} = q_i + hM^{-1}p_{i+1/2}$
    $p_{i+1} = p_{i+1/2} - \frac{h}{2}\nabla V_{\mathsf{fast}}(q_{i+1})$
end
$q_{n+1} = q_k$
$p_{n+1} = p_k - \frac{kh}{2}\nabla V_{\mathsf{slow}}(q_{n+1})$
```

LIN:

Langevin Dynamics with Taylor series forces.

$$
\begin{aligned}
\dot{q} &= M^{-1}p \\
\dot{p} &= [-\nabla V(q_0) - H(q - q_0) - \ldots] - \gamma p + r
\end{aligned}
$$

Discretization:

LIN family of methods

$\boxed{\text{solve}}$ the linearized system over $[0, h]$, with initial conditions $q^0_{\text{ref}} = q_n$, $p^0_{\text{ref}} = p_n$

$$
\begin{aligned}
\dot{q}_{\text{ref}} &= M^{-1}p_{\text{ref}} \\
\dot{p}_{\text{ref}} &= -\nabla V(q_0) - H(q_{\text{ref}} - q_0) - \gamma p_{\text{ef}} + r
\end{aligned}
$$

$\boxed{\textbf{integrate}}$ the residual with timestep h

$$
\begin{aligned}
\overset{\cdot}{(q - q_{\text{ref}})} &= M^{-1}(p - p_{\text{ref}}) \\
\overset{\cdot}{(p - p_{\text{ref}})} &= -\nabla V(q) + \nabla V(q_0) + H(q_{\text{ref}} - q_0) \\
&\quad -\gamma(p - p_{\text{ref}})
\end{aligned}
$$

VERLET - type integrators
dominate
MACROMOLECULAR (BIO-) DYNAMICS

* one-step integrator
(one evaluation of forces per timestep)

* relatively high order
("feels" the curvature)

* time-reversible, simplectic
(good long-time behavior)

"STABLE INTEGRATION"

in biomolecular dynamics

1) $$\Delta E = \frac{1}{N} \sum_i \left| \frac{E_{initial} - E_i}{E_{initial}} \right|$$

$$\Delta E \leqslant 0.001 - 0.003$$

2) $$R = \frac{\Delta E_{rms}}{\Delta KE_{rms}}$$

$$R \leqslant 0.01$$

Rule of thumb: $\Delta t = T/20$

SHAKE

(Ryckaert, Ciccotti, Berendsen, J. Comp. Phys., 23, 327 (1977))

- BOND LENGTH CONSTRAINTS - small effect on long-time dynamics

- ANGLE CONSTRAINTS - reduce mobility of the system

BPTI - 58 res.

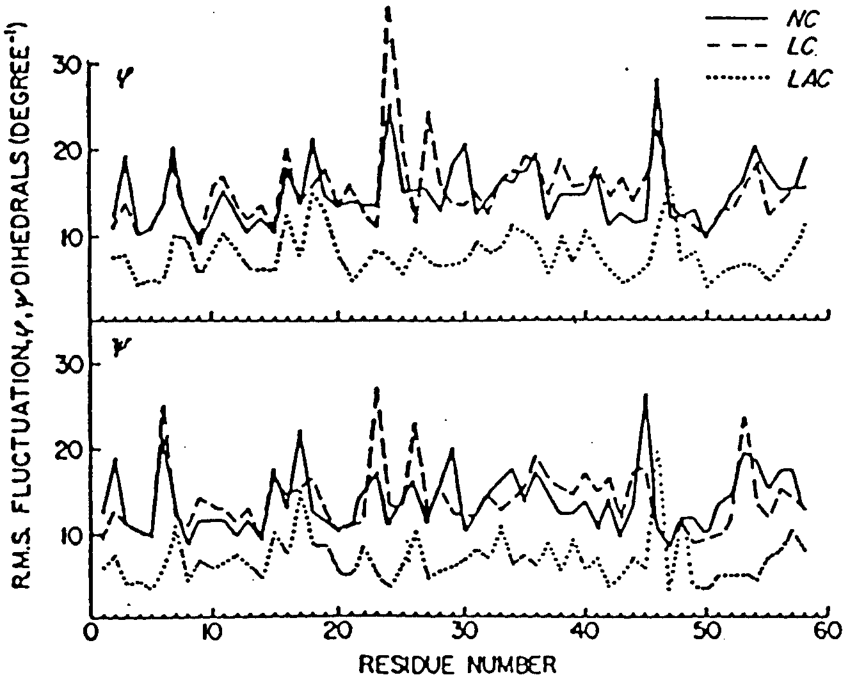


Figure 7. Root-mean-square fluctuations of the dihedrals angles ϕ,ψ over 25 "ps": (—) NC run; (---) LC run; (···) LAC run.

(van Gunsteren, Karplus, Macromolecules 15, 1528 (1982))

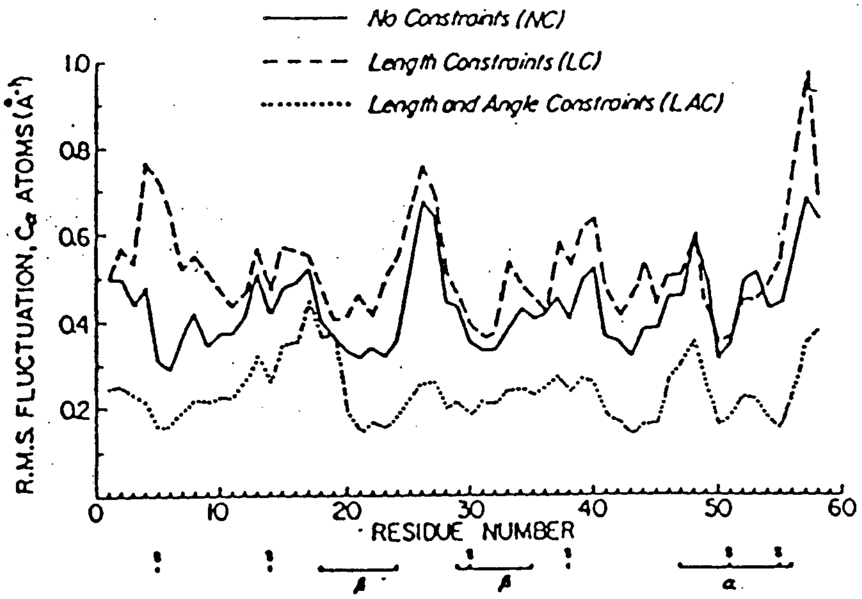


Figure 3. Root-mean-square fluctuations of C_α atoms over 25 "ps": (—) NC run; (---) LC run; (···) LAC run.

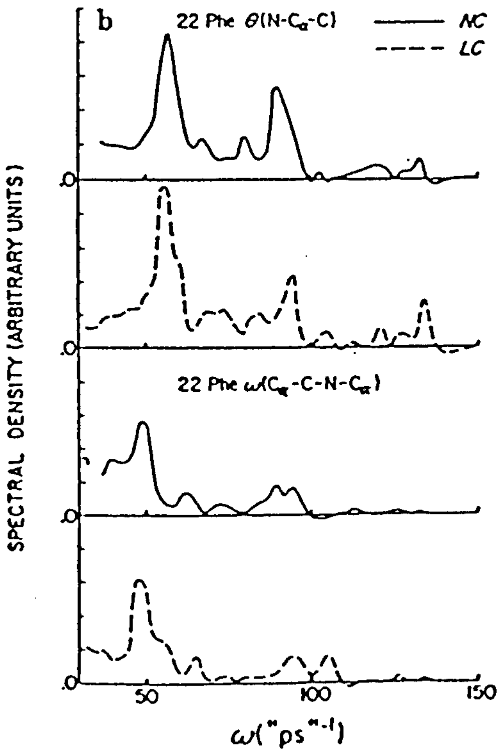


Figure 6. Bond-angle and dihedral angle autocorrelation and spectral density functions: (—) NC run; (---) LC run. (a) Autocorrelation function; (b) spectral density.

TABLE 3: Combined Stiff/Soft, Internal/External, and Short/Long-Range Nonbonded Force Decomposition. Comparison of Energy Conservation and Associated CPU Times Spent in the Various Force Routines for Velocity Verlet ($n_1 = n_2 = n_3 = 1$) and r-RESPA Using the Propagator Given by eq 38 and eq 43[a]

Δt	$\delta\tau_1$	n_1	n_2	n_3	$\log(\Delta E)$	R	T_{stretch}	T_{bend}	T_{torsion}	$T_{\mathrm{nonbonded}}$
0.25	0.25	1	1	1	−3.7073	0.0022	30.7	186.0	399.7	10085.9
0.50	0.50	1	1	1	−3.0123	0.0087	15.5	93.3	201.2	5073.4
1.00	1.00	1	1	1	−2.2388	0.0398	7.7	47.9	102.8	2579.6
2.00	2.00	1	1	1	−1.0475	0.1981	4.2	23.6	52.5	1326.7
3.00	0.25	1	6	2	−2.9880	0.0102	31.9	189.1	407.6	1090.0
3.00	0.25	2	3	2	−2.9995	0.0102	32.1	92.3	198.7	1062.9
3.00	0.50	1	3	2	−2.7944	0.0127	15.6	93.6	201.0	1078.2

[a] $\log(\Delta E)$ and R are given by eq 48 and eq 49. Δt and $\delta\tau_1$ are in fentoseconds. Here $r_c = 6.0$ Å and $\Delta r = 2.0$ Å for all r-RESPA cases considered. T_{stretch}, T_{bend}, T_{torsion}, and $T_{\mathrm{nonbonded}}$ are the CPU times in seconds spent in the stretch, bend, torsion, and nonbonded force routines, respectively. CPU time spent calculating r-RESPA neighbor lists is included in $T_{\mathrm{nonbonded}}$.

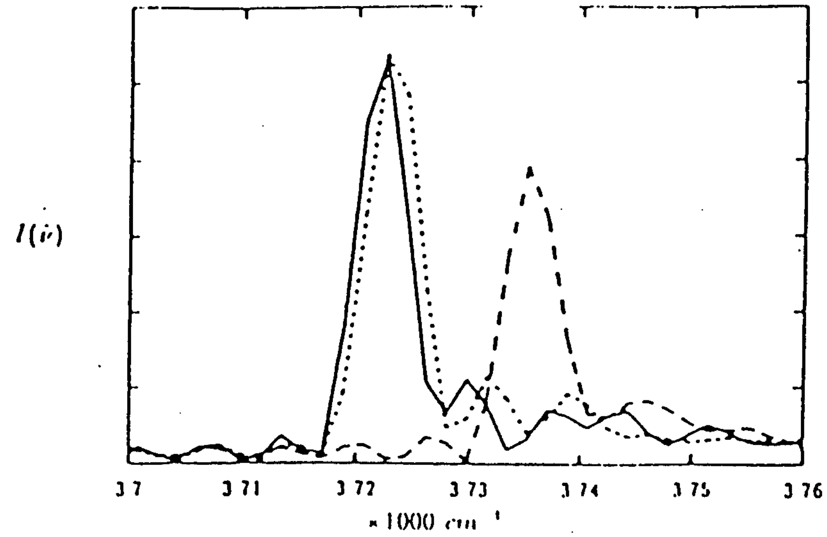


Figure 6. Detail of spectral density $I(\tilde{\nu})$ of eq 52 as a function of wavenumber for "Exact" (i.e. velocity Verlet Δt = 0.25 fs) (solid line), velocity Verlet Δt = 0.5 fs (dashed line), and r-RESPA using the propagator given by eq 38 and eq 43 with n_1 = 1, n_2 = 6, n_3 = 2, and δt_1 = 0.25 fs (dotted line). Intensities are in arbitrary units.

r-RESPA (Tuckerman, Berne, Martyna, JCP 97, 1990 (1992))

EFFICIENCY of the method depends on
the optimal split of the forces

(Humphreys, Friesner, Berne, JPC 98, 6885 (1994))
crambin - 46 res.

$$\begin{tikzcd} V = V_b + V_\theta + V_\varphi + V_{nb} \arrow[d] \arrow[dr] & \\ V_{nb}(1) & V_{nb}(2) \end{tikzcd}$$

$$\underbrace{V_b}_{\tau_1} \quad \underbrace{V_\theta + V_\varphi}_{n_1\tau_1} \quad \underbrace{V_{nb}(1)}_{n_1 \cdot n_2 \tau_1} \quad \underbrace{V_{nb}(2)}_{n_1 n_2 n_3 \tau_1 = \Delta t}$$

Watanabe, Karplus, J. Phys. Chem. 99, 5680 (1995)

Single r - RESPA

$$V = \underbrace{V_b + V_\theta} + \underbrace{V_\varphi + V_{nb}}$$

Double r-RESPA

$$V = V_b + V_\theta + V_\varphi + V_{nb}$$

$$\underbrace{\qquad\qquad\qquad} \quad \underbrace{V_{nb}(H)} + \underbrace{V_{nb}(\text{rest})}$$

BPTI

TABLE 10: Average Total Energies, Fluctuations of Total and Kinetic Energies, and CPU Ratios for the BPTI Molecule

model	$\langle E_{total} \rangle$ (kcal/mol)	ΔE_{total} (kcal/mol)	ΔE_{kin} (kcal/mol)	$\log(\Delta E_{rms})^a$	CPU ratio[b]	
		Standard				
1	1134.55	0.127	15.40	−2.08	1.00	0.5 fs
2	1134.48	0.545	15.76	−1.46	0.52	1.0 fs
		SHAKE				
	1061.12	0.844	15.81	−1.26	0.28	2.0 fs
		Single r-RESPA				
1	1134.71	0.290	15.40	−1.73	0.28	0.5fs - 1.0fs
2	1135.21	0.498	15.38	−1.49	0.22	0.5fs - 2.5fs
3[c]					0.21	B,A rest
		Double r-RESPA				
1	1134.41	0.562	15.20	−1.43	0.26	0.5 - 2.0 - 4.0
2	1135.36	0.662	15.48	−1.37	0.22	0.5 - 2.5 - 5.0
3[c]					0.19	internal nb_H nb

[a] See eq 18. All values calculated over 10 ps of production run. [b] Each CPU ratio was calculated with CPU time (method)/CPU time (standard 1). [c] Total energies of sr-RESPA 3 and dr-RESPA 5 are not conserved.

LIN (Zhang, Schlick, J. Comp. Chem. 14, 1212 (1993))

Langevin dynamics - total energy fluctuate

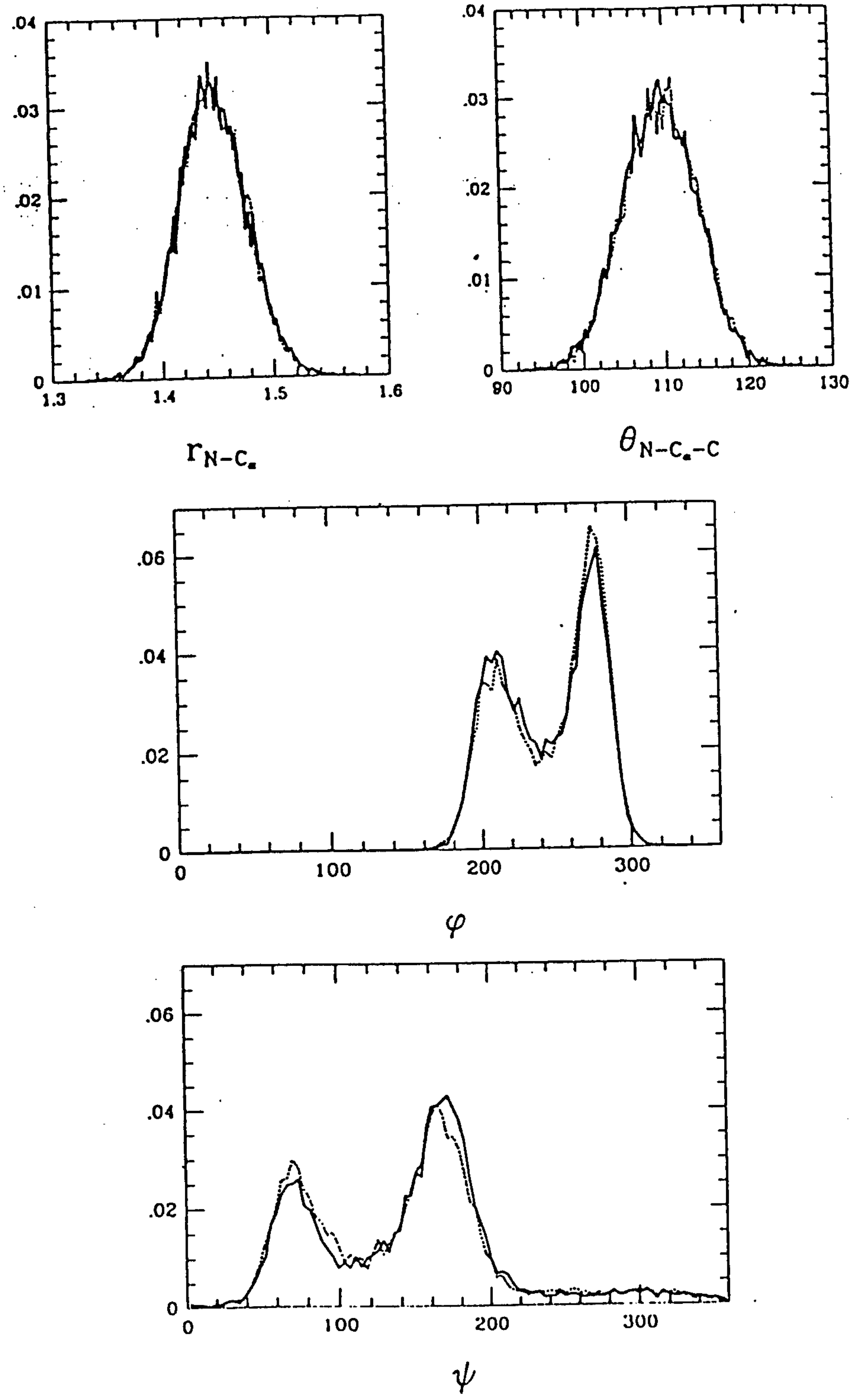
Averages for Alanine Dipeptide, LIN (Δt = 30 fs) vs. Explicit ($\Delta t = 0.5$ fs) Langevin Trajectories 2 ns

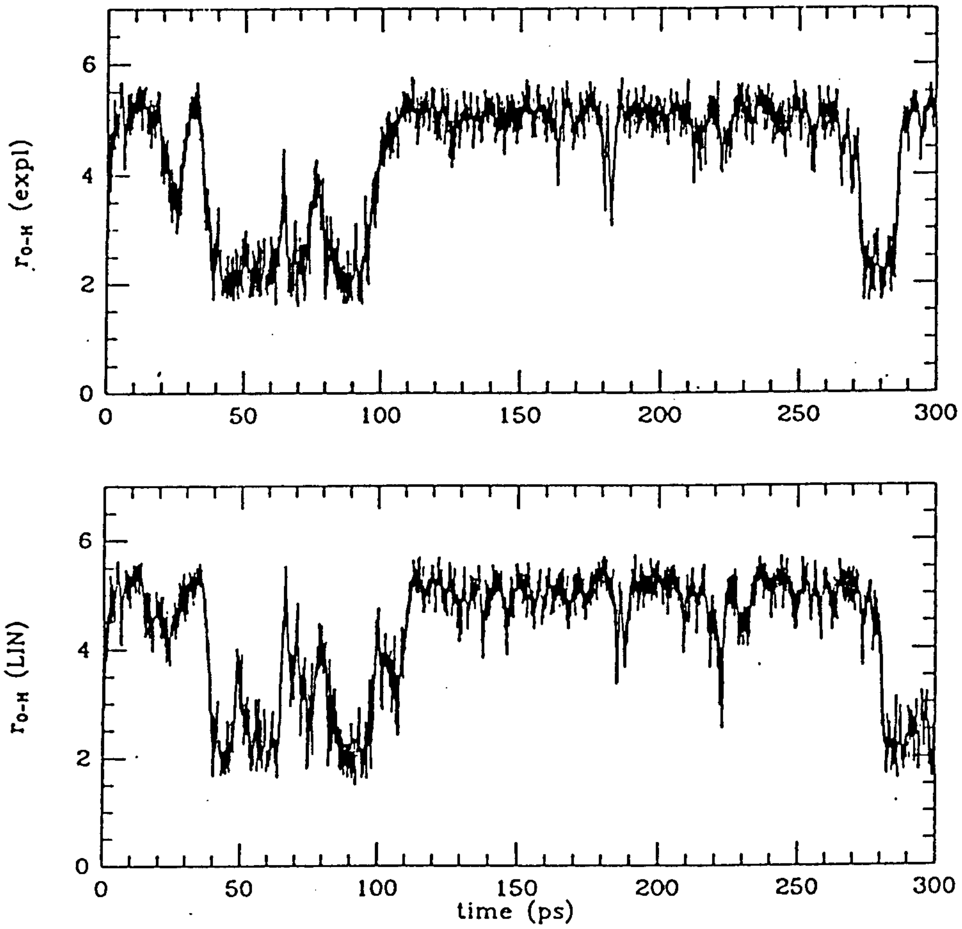
	mean, ex.	var., ex.	mean, LIN	var., LIN
E[a]	35.0	4.8	36.8	5.4
E_k	19.7	3.4	20.6	3.6
E_p	15.4	3.3	16.6	3.8
ϕ[b]	−116.0	32.9	−113.2	33.0
ψ	64.6	90.2	66.9	86.2
r_{CN}[c]	1.344	.028	1.345	.028
r_{NC_α}	1.446	.030	1.447	.030
$r_{C_\alpha C}$	1.520	.034	1.520	.034
θ_{CNC_α}	123.4	3.3	123.2	3.3
$\theta_{NC_\alpha C}$	109.3	4.4	109.5	4.5
$\theta_{C_\alpha CN}$	117.0	2.8	117.0	2.9

[a] Energy is given in kcal/mol

[b] Angles are measured in degrees

[c] Bond lengths are given in Å


.04
.03
.02
.01
0
1.3
1.4
1.5
1.6
r_{N-C_α}
.04
.03
.02
.01
0
90
100
110
120
130
$\theta_{N-C_\alpha-C}$
.06
.04
.02
0
0
100
200
300
φ
.06
.04
.02
0
0
100
200
300
ψ

r_{O-H} (expl)
6
4
2
0
0
50
100
150
200
250
300
r_{O-H} (LIN)
6
4
2
0
0
50
100
150
200
250
300
time (ps)

COMPARISON OF LONG TIMESTEP APPROACHES

METHOD	Approximation	Δt (fs)
VERLET		0.5
SHAKE	constrains	2.0
REDUCED VARIABLE CONF. SAMPL.		2.0
SUBSTRUCTURING		
RESPA	multiple timestep	3.0 5.0
LIN	separating framework	29.0

BROWNIAN DYNAMICS	constrains substructuring	10^3 40.0
ESSENTIAL SUBSPACE	normal modes local modes $< \Delta r_i \Delta r_j >$	16.7

Efficent Dense Hessian Computations in Molecular Minimization

Christophe G. Lambert

Duke University
Department of Computer Science
lambert@cs.duke.edu

Overview of Presentation

- Introduction and Motivation
- Newton-Based Minimization and the Hessian
- Need for Hessian-Vector Product in Iterative Matrix Solve
- Goal: Describe Product in Terms of N-body Problem
- The Hessian Matrix Structure
- Decompose Product into Off-Diagonal and Diagonal Parts
- Discussion of Complete Computation
- Ongoing Work

Introduction and Motivation

- Diverse applications of Potential Energy Minimization
  - NMR and X-Ray structure refinement
  - Molecular modeling
  - **Long time step dynamics (Schlick)**
- Newton Methods
  - Quadratic convergence near minimum
  - Large computational overhead
    - $O(n^2)$ space
    - $O(n^3)$ time with direct matrix solve
    - $O(n^2)$ time per iteration within iterative matrix solve
- Goal is to reduce overhead of Newton method
  - $O(n)$ space
  - $O(n)$ time per iteration within iterative matrix solve
  - Include all electrostatic interactions (no cutoffs)

Newton-Based Minimization

Goal: minimize $E(z)$, $E : \Re^n \rightarrow \Re$.

Expand the function $E(z + z^*)$ about z^* as:

$$E(z+z^*) = E(z^*) + \sum_{i=1}^{n} \frac{\partial E(z^*)}{\partial z_i} z_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 E(z^*)}{\partial z_i \partial z_j} z_i z_j + O(||z^*||^3).$$

Set gradient of quadratic approximation to zero:

$$\frac{\partial E(z+z^*)}{\partial z_k} = \frac{\partial E(z^*)}{\partial z_k} + \sum_{j=1}^{n} \frac{\partial^2 E(z^*)}{\partial z_k \partial z_j} z_j = 0, \, \forall \, k = 1 \ldots n.$$

Express this in matrix form:

$$\begin{bmatrix} \frac{\partial E(z^*)}{\partial z_1} \\ \frac{\partial E(z^*)}{\partial z_2} \\ \vdots \\ \frac{\partial E(z^*)}{\partial z_n} \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 E(z^*)}{\partial z_1 z_1} & \frac{\partial^2 E(z^*)}{\partial z_1 z_2} & \cdots & \frac{\partial^2 E(z^*)}{\partial z_1 z_n} \\ \frac{\partial^2 E(z^*)}{\partial z_2 z_1} & \frac{\partial^2 E(z^*)}{\partial z_2 z_2} & \cdots & \frac{\partial^2 E(z^*)}{\partial z_2 z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(z^*)}{\partial z_n z_1} & \frac{\partial^2 E(z^*)}{\partial z_n z_2} & \cdots & \frac{\partial^2 E(z^*)}{\partial z_n z_n} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = 0,$$

or

$$H(z^*)z = -g(z^*).$$

Newton Minimization Algorithm

1. Start with initial guess for the minimum, z^*.
2. Compute the gradient g and Hessian H at z^*.
3. **Solve the equation $Hz = -g$.**
4. Set z^* to $z + z^*$. If not converged, goto step 2.

Iterative Matrix Solve

- Preconditioned Conjugate Gradients
- Need Fast Hessian-Vector Product
- Goal is $O(n)$ Product

$\exists\ O(n)$ Algorithm to Solve N-Body Problem:

$$-\vec{F}_x = \begin{bmatrix} q_1\frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}$$

$$= diag\left(q_i\frac{\partial}{x_i}\right) \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix},$$

where

$$\Lambda_{i,j} = \frac{1}{r} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}$$

$$\Phi_i = \sum_{j \neq i} q_j \Lambda_{i,j}.$$

Goal: Describe Hessian Matrix-Vector Product in Terms of N-Body Problem

Hessian Matrix Structure

$$E = \sum_{i=1}^{n} \sum_{j>i} \frac{q_i q_j}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}.$$

$$\begin{bmatrix}
\begin{bmatrix} \frac{\partial^2 E}{\partial x_1 \partial x_1} & \frac{\partial^2 E}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_n} \\ \frac{\partial^2 E}{\partial x_2 \partial x_1} & \frac{\partial^2 E}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial x_1} & \frac{\partial^2 E}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_n \partial x_n} \end{bmatrix} &
\begin{bmatrix} \frac{\partial^2 E}{\partial x_1 \partial y_1} & \frac{\partial^2 E}{\partial x_1 \partial y_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial y_n} \\ \frac{\partial^2 E}{\partial x_2 \partial y_1} & \frac{\partial^2 E}{\partial x_2 \partial y_2} & \cdots & \frac{\partial^2 E}{\partial x_2 \partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial y_1} & \frac{\partial^2 E}{\partial x_n \partial y_2} & \cdots & \frac{\partial^2 E}{\partial x_n \partial y_n} \end{bmatrix} &
\begin{bmatrix} \frac{\partial^2 E}{\partial x_1 \partial z_1} & \frac{\partial^2 E}{\partial x_1 \partial z_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial z_n} \\ \frac{\partial^2 E}{\partial x_2 \partial z_1} & \frac{\partial^2 E}{\partial x_2 \partial z_2} & \cdots & \frac{\partial^2 E}{\partial x_2 \partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial z_1} & \frac{\partial^2 E}{\partial x_n \partial z_2} & \cdots & \frac{\partial^2 E}{\partial x_n \partial z_n} \end{bmatrix} \\
\begin{bmatrix} \frac{\partial^2 E}{\partial y_1 \partial x_1} & \frac{\partial^2 E}{\partial y_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial y_1 \partial x_n} \\ \frac{\partial^2 E}{\partial y_2 \partial x_1} & \frac{\partial^2 E}{\partial y_2 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial y_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial y_n \partial x_1} & \frac{\partial^2 E}{\partial y_n \partial x_2} & \cdots & \frac{\partial^2 E}{\partial y_n \partial x_n} \end{bmatrix} &
\begin{bmatrix} \frac{\partial^2 E}{\partial y_1 \partial y_1} & \frac{\partial^2 E}{\partial y_1 \partial y_2} & \cdots & \frac{\partial^2 E}{\partial y_1 \partial y_n} \\ \frac{\partial^2 E}{\partial y_2 \partial y_1} & \frac{\partial^2 E}{\partial y_2 \partial y_2} & \cdots & \frac{\partial^2 E}{\partial y_2 \partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial y_n \partial y_1} & \frac{\partial^2 E}{\partial y_n \partial y_2} & \cdots & \frac{\partial^2 E}{\partial y_n \partial y_n} \end{bmatrix} &
\begin{bmatrix} \frac{\partial^2 E}{\partial y_1 \partial z_1} & \frac{\partial^2 E}{\partial y_1 \partial z_2} & \cdots & \frac{\partial^2 E}{\partial y_1 \partial z_n} \\ \frac{\partial^2 E}{\partial y_2 \partial z_1} & \frac{\partial^2 E}{\partial y_2 \partial z_2} & \cdots & \frac{\partial^2 E}{\partial y_2 \partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial y_n \partial z_1} & \frac{\partial^2 E}{\partial y_n \partial z_2} & \cdots & \frac{\partial^2 E}{\partial y_n \partial z_n} \end{bmatrix} \\
\begin{bmatrix} \frac{\partial^2 E}{\partial z_1 \partial x_1} & \frac{\partial^2 E}{\partial z_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial z_1 \partial x_n} \\ \frac{\partial^2 E}{\partial z_2 \partial x_1} & \frac{\partial^2 E}{\partial z_2 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial z_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial z_n \partial x_1} & \frac{\partial^2 E}{\partial z_n \partial x_2} & \cdots & \frac{\partial^2 E}{\partial z_n \partial x_n} \end{bmatrix} &
\begin{bmatrix} \frac{\partial^2 E}{\partial z_1 \partial y_1} & \frac{\partial^2 E}{\partial z_1 \partial y_2} & \cdots & \frac{\partial^2 E}{\partial z_1 \partial y_n} \\ \frac{\partial^2 E}{\partial z_2 \partial y_1} & \frac{\partial^2 E}{\partial z_2 \partial y_2} & \cdots & \frac{\partial^2 E}{\partial z_2 \partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial z_n \partial y_1} & \frac{\partial^2 E}{\partial z_n \partial y_2} & \cdots & \frac{\partial^2 E}{\partial z_n \partial y_n} \end{bmatrix} &
\begin{bmatrix} \frac{\partial^2 E}{\partial z_1 \partial z_1} & \frac{\partial^2 E}{\partial z_1 \partial z_2} & \cdots & \frac{\partial^2 E}{\partial z_1 \partial z_n} \\ \frac{\partial^2 E}{\partial z_2 \partial z_1} & \frac{\partial^2 E}{\partial z_2 \partial z_2} & \cdots & \frac{\partial^2 E}{\partial z_2 \partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial z_n \partial z_1} & \frac{\partial^2 E}{\partial z_n \partial z_2} & \cdots & \frac{\partial^2 E}{\partial z_n \partial z_n} \end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \vdots \\ \overline{x}_n \end{bmatrix} \\
\begin{bmatrix} \overline{y}_1 \\ \overline{y}_2 \\ \vdots \\ \overline{y}_n \end{bmatrix} \\
\begin{bmatrix} \overline{z}_1 \\ \overline{z}_2 \\ \vdots \\ \overline{z}_n \end{bmatrix}
\end{bmatrix}$$

Off-Diagonal Components of Product

Consider off-diagonal parts of upper left subproduct:

$$\begin{bmatrix} 0 & \frac{\partial^2 E}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_n} \\ \frac{\partial^2 E}{\partial x_2 \partial x_1} & 0 & \cdots & \frac{\partial^2 E}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial x_1} & \frac{\partial^2 E}{\partial x_n \partial x_2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \vdots \\ \overline{x}_n \end{bmatrix}$$

Let:

$$\Lambda_{i,j} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}.$$

We can rewrite the off-diagonal part of this matrix as:

$$\begin{bmatrix} q_1 \frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2 \frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \frac{\partial}{x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \overline{x}_1 \frac{\partial}{x_1} \\ q_2 \overline{x}_2 \frac{\partial}{x_2} \\ \vdots \\ q_n \overline{x}_n \frac{\partial}{x_n} \end{bmatrix}.$$

Let:

$$Q_{x_i} = q_i \overline{x_i}$$

$$\Lambda_{x_{i,j}} = \frac{\partial}{x_j} \Lambda_{i,j} = \frac{x_i - x_j}{\left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right)^{\frac{3}{2}}}$$

Rewrite product as:

$$\begin{bmatrix} q_1 \frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2 \frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \frac{\partial}{x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{x_{1,2}} & \cdots & \Lambda_{x_{1,n}} \\ \Lambda_{x_{2,1}} & 0 & \cdots & \Lambda_{x_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{x_{n,1}} & \Lambda_{x_{n,2}} & \cdots & 0 \end{bmatrix} \begin{bmatrix} Q_{x_1} \\ Q_{x_2} \\ \vdots \\ Q_{x_n} \end{bmatrix}.$$

$\exists\ O(n)$ Algorithm to Solve N-Body Problem:

$$
\begin{aligned}
-\vec{F}_x &= \begin{bmatrix} q_1\frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{x_n} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \\
&= diag\left(q_i\frac{\partial}{x_i}\right) \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix},
\end{aligned}
$$

where

$$\Lambda_{i,j} = \frac{1}{r} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}$$

$$\Phi_i = \sum_{j \neq i} q_j \Lambda_{i,j}.$$

Diagonal Components of Product

Diagonal part of upper left sub-product:

$$\begin{bmatrix} \frac{\partial^2 E}{\partial x_1 \partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial^2 E}{\partial x_2 \partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^2 E}{\partial x_n \partial x_n} \end{bmatrix} \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \vdots \\ \overline{x}_n \end{bmatrix}$$

is equivalent to:

$$\begin{bmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial {x_1}^2} \sum_{j \neq 1} q_j \Lambda_{1,j} & 0 & \cdots & 0 \\ 0 & \frac{\partial^2}{\partial {x_2}^2} \sum_{j \neq 2} q_j \Lambda_{2,j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^2}{\partial {x_n}^2} \sum_{j \neq n} q_j \Lambda_{n,j} \end{bmatrix} \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \vdots \\ \overline{x}_n \end{bmatrix},$$

which can be rewritten as:

$$\begin{bmatrix} q_1 \overline{x}_1 \frac{\partial^2}{\partial {x_1}^2} & 0 & \cdots & 0 \\ 0 & q_2 \overline{x}_2 \frac{\partial^2}{\partial {x_2}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \overline{x}_n \frac{\partial^2}{\partial {x_n}^2} \end{bmatrix} \begin{bmatrix} 0 & \Lambda_{1,2} & \cdots & \Lambda_{1,n} \\ \Lambda_{2,1} & 0 & \cdots & \Lambda_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{n,1} & \Lambda_{n,2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix},$$

or

$$\begin{bmatrix} q_1 \overline{x}_1 \frac{\partial^2}{\partial {x_1}^2} & 0 & \cdots & 0 \\ 0 & q_2 \overline{x}_2 \frac{\partial^2}{\partial {x_2}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n \overline{x}_n \frac{\partial^2}{\partial {x_n}^2} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix}.$$

The Complete Computation

$$
\begin{bmatrix}
\begin{bmatrix} q_1\frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{x_n} \end{bmatrix} &
\begin{bmatrix} q_1\frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{x_n} \end{bmatrix} &
\begin{bmatrix} q_1\frac{\partial}{x_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{x_n} \end{bmatrix} \\
\begin{bmatrix} q_1\frac{\partial}{y_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{y_n} \end{bmatrix} &
\begin{bmatrix} q_1\frac{\partial}{y_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{y_n} \end{bmatrix} &
\begin{bmatrix} q_1\frac{\partial}{y_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{y_n} \end{bmatrix} \\
\begin{bmatrix} q_1\frac{\partial}{z_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{z_n} \end{bmatrix} &
\begin{bmatrix} q_1\frac{\partial}{z_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{z_n} \end{bmatrix} &
\begin{bmatrix} q_1\frac{\partial}{z_1} & 0 & \cdots & 0 \\ 0 & q_2\frac{\partial}{z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_n\frac{\partial}{z_n} \end{bmatrix}
\end{bmatrix} \times
$$

$$
\begin{bmatrix}
\begin{bmatrix} 0 & \Lambda_{x_{1,2}} & \cdots & \Lambda_{x_{1,n}} \\ \Lambda_{x_{2,1}} & 0 & \cdots & \Lambda_{x_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{x_{n,1}} & \Lambda_{x_{n,2}} & \cdots & 0 \end{bmatrix} &
\begin{bmatrix} 0 & \Lambda_{y_{1,2}} & \cdots & \Lambda_{y_{1,n}} \\ \Lambda_{y_{2,1}} & 0 & \cdots & \Lambda_{y_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{y_{n,1}} & \Lambda_{y_{n,2}} & \cdots & 0 \end{bmatrix} &
\begin{bmatrix} 0 & \Lambda_{z_{1,2}} & \cdots & \Lambda_{z_{1,n}} \\ \Lambda_{z_{2,1}} & 0 & \cdots & \Lambda_{z_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{z_{n,1}} & \Lambda_{z_{n,2}} & \cdots & 0 \end{bmatrix} \\
\begin{bmatrix} 0 & \Lambda_{x_{1,2}} & \cdots & \Lambda_{x_{1,n}} \\ \Lambda_{x_{2,1}} & 0 & \cdots & \Lambda_{x_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{x_{n,1}} & \Lambda_{x_{n,2}} & \cdots & 0 \end{bmatrix} &
\begin{bmatrix} 0 & \Lambda_{y_{1,2}} & \cdots & \Lambda_{y_{1,n}} \\ \Lambda_{y_{2,1}} & 0 & \cdots & \Lambda_{y_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{y_{n,1}} & \Lambda_{y_{n,2}} & \cdots & 0 \end{bmatrix} &
\begin{bmatrix} 0 & \Lambda_{z_{1,2}} & \cdots & \Lambda_{z_{1,n}} \\ \Lambda_{z_{2,1}} & 0 & \cdots & \Lambda_{z_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{z_{n,1}} & \Lambda_{z_{n,2}} & \cdots & 0 \end{bmatrix} \\
\begin{bmatrix} 0 & \Lambda_{x_{1,2}} & \cdots & \Lambda_{x_{1,n}} \\ \Lambda_{x_{2,1}} & 0 & \cdots & \Lambda_{x_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{x_{n,1}} & \Lambda_{x_{n,2}} & \cdots & 0 \end{bmatrix} &
\begin{bmatrix} 0 & \Lambda_{y_{1,2}} & \cdots & \Lambda_{y_{1,n}} \\ \Lambda_{y_{2,1}} & 0 & \cdots & \Lambda_{y_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{y_{n,1}} & \Lambda_{y_{n,2}} & \cdots & 0 \end{bmatrix} &
\begin{bmatrix} 0 & \Lambda_{z_{1,2}} & \cdots & \Lambda_{z_{1,n}} \\ \Lambda_{z_{2,1}} & 0 & \cdots & \Lambda_{z_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{z_{n,1}} & \Lambda_{z_{n,2}} & \cdots & 0 \end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\begin{bmatrix} q_1\overline{x}_1 \\ q_2\overline{x}_2 \\ \vdots \\ q_n\overline{x}_n \end{bmatrix} \\
\begin{bmatrix} q_1\overline{y}_1 \\ q_2\overline{y}_2 \\ \vdots \\ q_n\overline{y}_n \end{bmatrix} \\
\begin{bmatrix} q_1\overline{z}_1 \\ q_2\overline{z}_2 \\ \vdots \\ q_n\overline{z}_n \end{bmatrix}
\end{bmatrix} +
$$

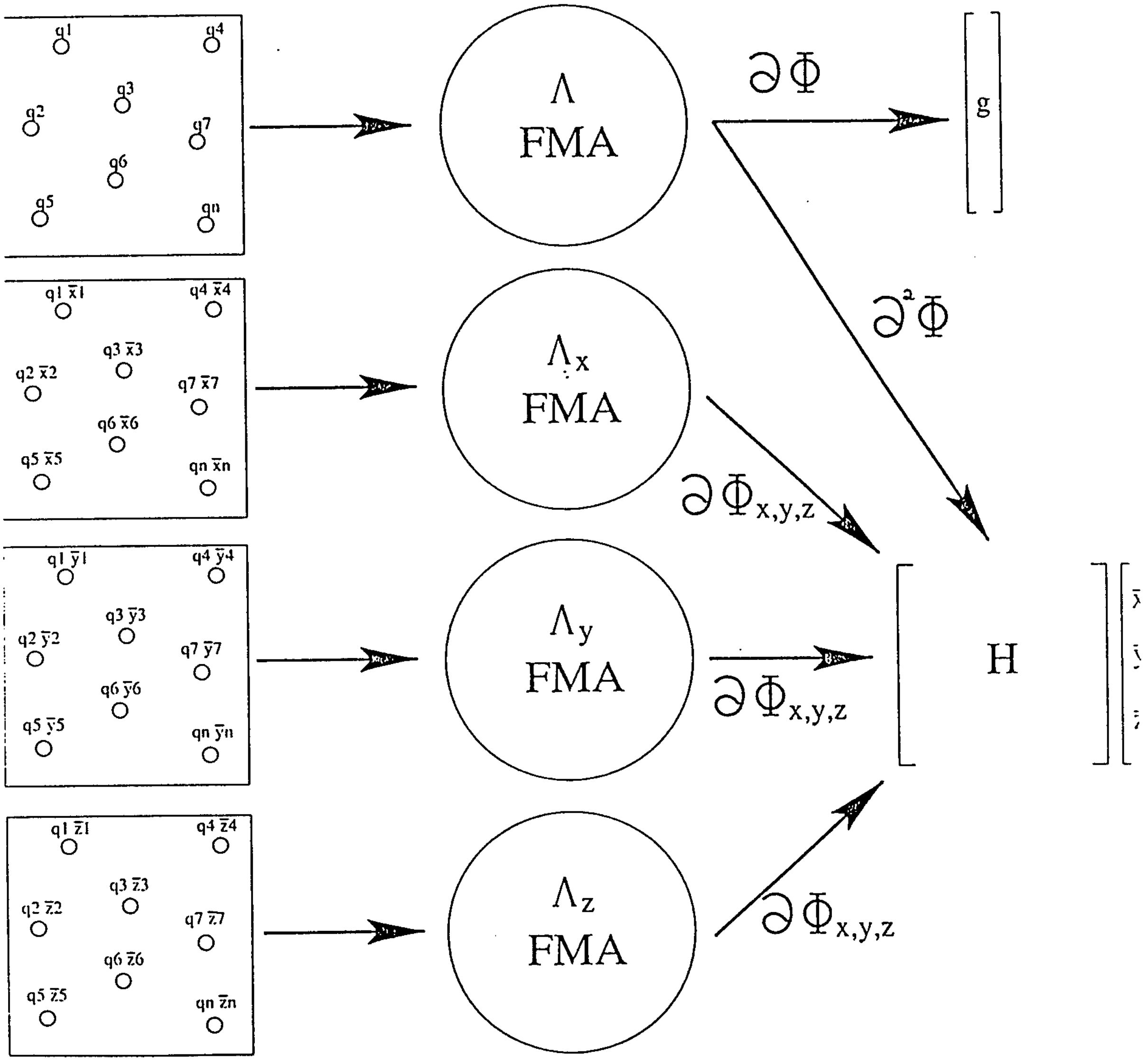
$$
\begin{bmatrix}
\begin{bmatrix} \frac{q_1\overline{x}_1\partial^2}{\partial x_1\partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{x}_2\partial^2}{\partial x_2\partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{x}_n\partial^2}{\partial x_n\partial x_n} \end{bmatrix} &
\begin{bmatrix} \frac{q_1\overline{y}_1\partial^2}{\partial x_1\partial y_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{y}_2\partial^2}{\partial x_2\partial y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{y}_n\partial^2}{\partial x_n\partial y_n} \end{bmatrix} &
\begin{bmatrix} \frac{q_1\overline{z}_1\partial^2}{\partial x_1\partial z_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{z}_2\partial^2}{\partial x_2\partial z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{z}_n\partial^2}{\partial x_n\partial z_n} \end{bmatrix} \\
\begin{bmatrix} \frac{q_1\overline{x}_1\partial^2}{\partial y_1\partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{x}_2\partial^2}{\partial y_2\partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{x}_n\partial^2}{\partial y_n\partial x_n} \end{bmatrix} &
\begin{bmatrix} \frac{q_1\overline{y}_1\partial^2}{\partial y_1\partial y_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{y}_2\partial^2}{\partial y_2\partial y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{y}_n\partial^2}{\partial y_n\partial y_n} \end{bmatrix} &
\begin{bmatrix} \frac{q_1\overline{z}_1\partial^2}{\partial y_1\partial z_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{z}_2\partial^2}{\partial y_2\partial z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{z}_n\partial^2}{\partial y_n\partial z_n} \end{bmatrix} \\
\begin{bmatrix} \frac{q_1\overline{x}_1\partial^2}{\partial z_1\partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{x}_2\partial^2}{\partial z_2\partial x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{x}_n\partial^2}{\partial z_n\partial x_n} \end{bmatrix} &
\begin{bmatrix} \frac{q_1\overline{y}_1\partial^2}{\partial z_1\partial y_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{y}_2\partial^2}{\partial z_2\partial y_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{y}_n\partial^2}{\partial z_n\partial y_n} \end{bmatrix} &
\begin{bmatrix} \frac{q_1\overline{z}_1\partial^2}{\partial z_1\partial z_1} & 0 & \cdots & 0 \\ 0 & \frac{q_2\overline{z}_2\partial^2}{\partial z_2\partial z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{q_n\overline{z}_n\partial^2}{\partial z_n\partial z_n} \end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix} \\
\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix} \\
\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix}
\end{bmatrix}
$$

More Compactly...

$$
\begin{bmatrix}
\left[q\frac{\partial}{x}\right] & \left[q\frac{\partial}{x}\right] & \left[q\frac{\partial}{x}\right] \\
\left[q\frac{\partial}{y}\right] & \left[q\frac{\partial}{y}\right] & \left[q\frac{\partial}{y}\right] \\
\left[q\frac{\partial}{z}\right] & \left[q\frac{\partial}{z}\right] & \left[q\frac{\partial}{z}\right]
\end{bmatrix}
\begin{bmatrix}
[\Lambda_x] & [\Lambda_y] & [\Lambda_z] \\
[\Lambda_x] & [\Lambda_y] & [\Lambda_z] \\
[\Lambda_x] & [\Lambda_y] & [\Lambda_z]
\end{bmatrix}
\begin{bmatrix}
[q\overline{x}] \\
[q\overline{y}] \\
[q\overline{z}]
\end{bmatrix}
+
$$

$$
\begin{bmatrix}
\left[q\overline{x}\,\frac{\partial^2}{\partial x\partial x}\right] & \left[q\overline{y}\,\frac{\partial^2}{\partial x\partial y}\right] & \left[q\overline{z}\,\frac{\partial^2}{\partial x\partial z}\right] \\
\left[q\overline{x}\,\frac{\partial^2}{\partial y\partial x}\right] & \left[q\overline{y}\,\frac{\partial^2}{\partial y\partial y}\right] & \left[q\overline{z}\,\frac{\partial^2}{\partial y\partial z}\right] \\
\left[q\overline{x}\,\frac{\partial^2}{\partial z\partial x}\right] & \left[q\overline{y}\,\frac{\partial^2}{\partial z\partial y}\right] & \left[q\overline{z}\,\frac{\partial^2}{\partial z\partial z}\right]
\end{bmatrix}
\begin{bmatrix}
[\Phi] \\
[\Phi] \\
[\Phi]
\end{bmatrix}
$$

Pictorially...



Ongoing Work

- Implement rudimentary $O(n \log n)$ tree-code without multipole translation operations
- Derive multipole translation operations for $\Lambda_x, \Lambda_y, \Lambda_z$ "potentials" for full-blown FMA
- Incorporate code into MD efforts at Schlick lab
- Parallel implementation

Acknowledgements

- Tamar Schlick (New York University)
- Xiaobai Sun (Duke University)
- John Board (Duke University)

Efficient Use of Fast Electrostatics in Molecular Dynamics

R. Skeel, M. Nelson, T. Bishop, K. Schulten
Theoretical Biophysics Group
Beckman Institute
University of Illinois

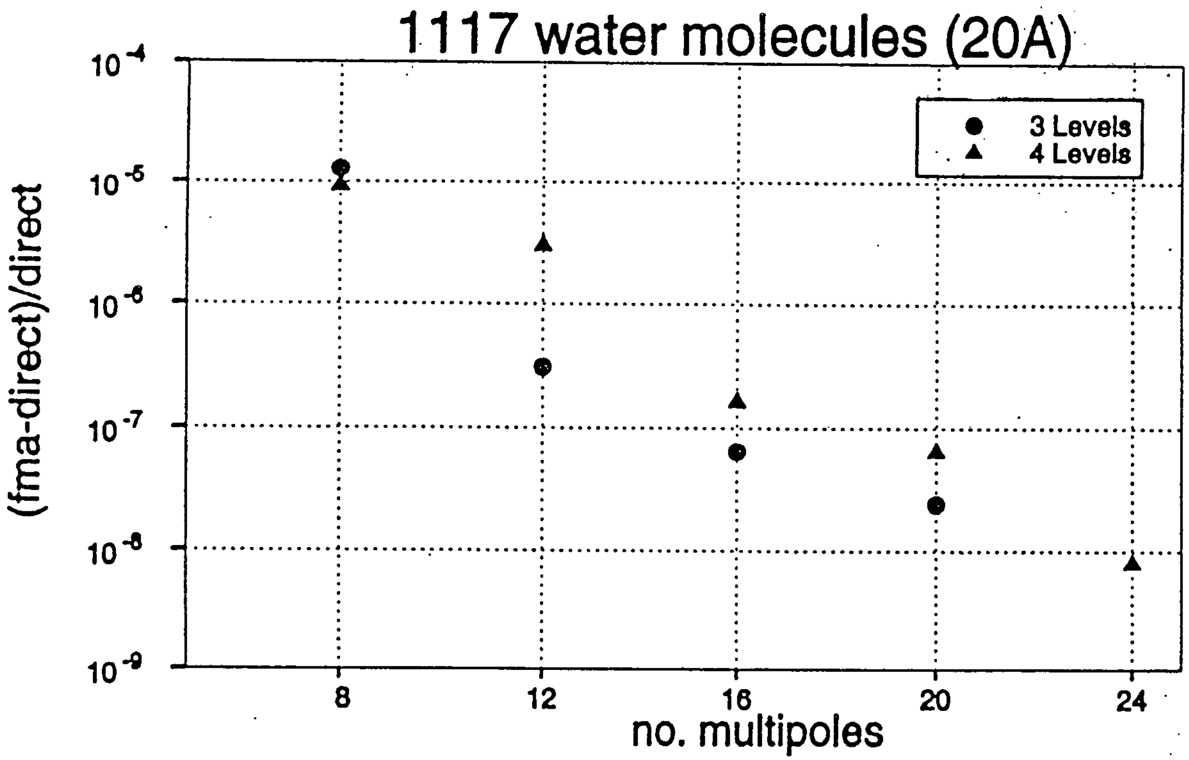
long-range electrostatics must often be included

e.g. protein-DNA, 30000 atoms

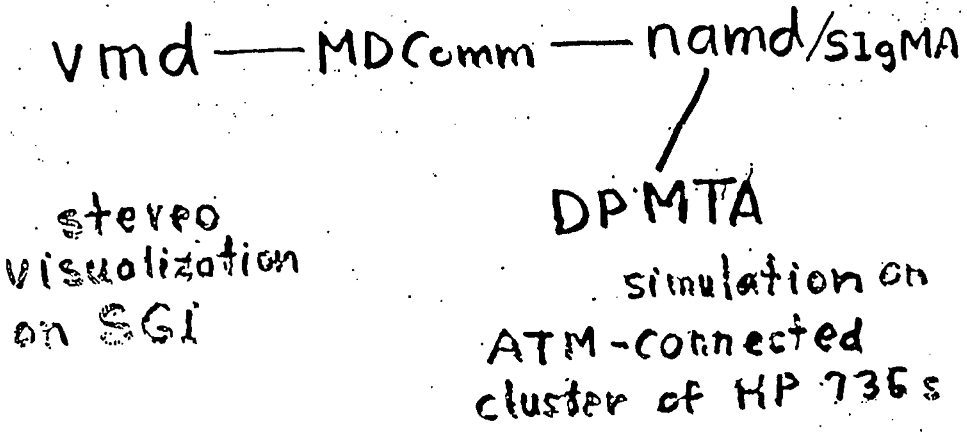
direct method	N^2
FMM to s. prec.	3000N

10 times the cost of using cutoffs

3


1117 water molecules (20A)
(fma-direct)/direct
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
8
12
16
20
24
no. multipoles
● 3 Levels
▲ 4 Levels

MDScope



vmd — MDComm — namd/SIgMA

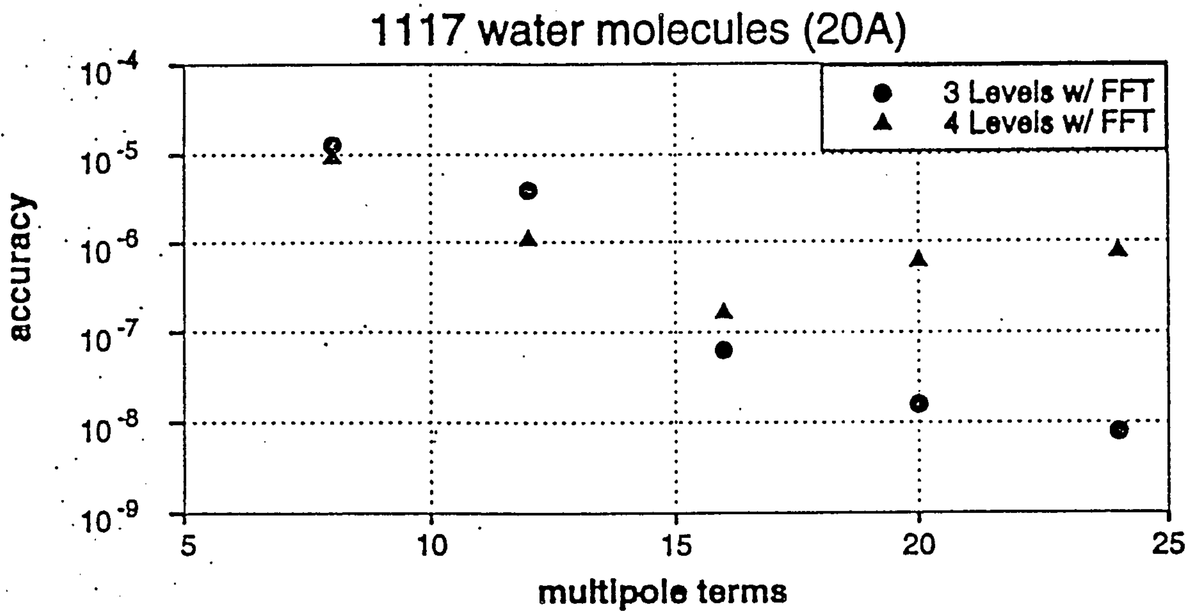
DPMTA

stereo visualization on SGI

simulation on ATM-connected cluster of HP 735s

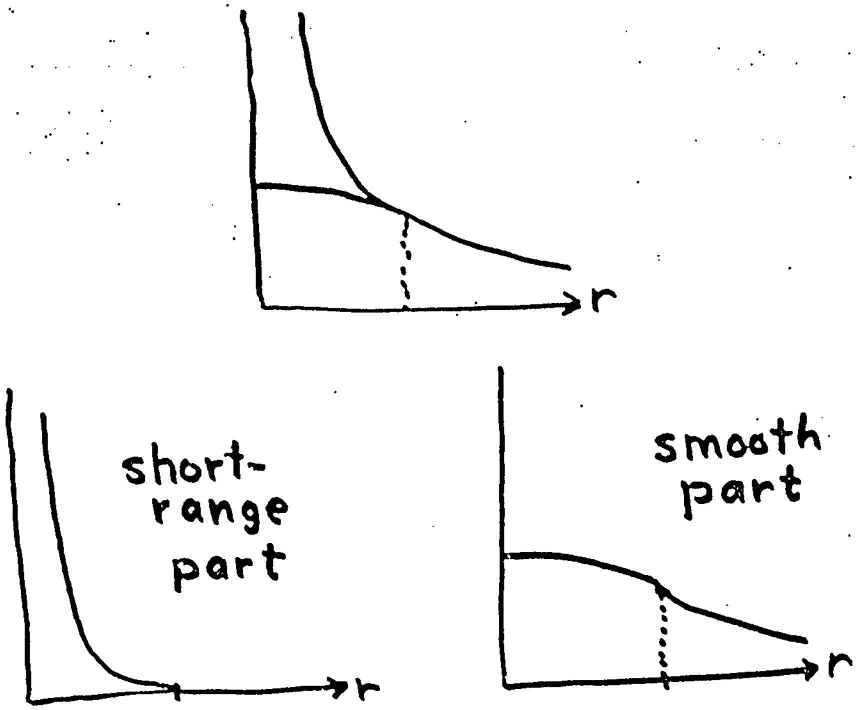
accuracy of DPMTA

4


1117 water molecules (20A)
accuracy
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
5
10
15
20
25
multipole terms
● 3 Levels w/ FFT
▲ 4 Levels w/ FFT

idea (partly A. Windemuth)

#1 artificial splitting of electrostatics

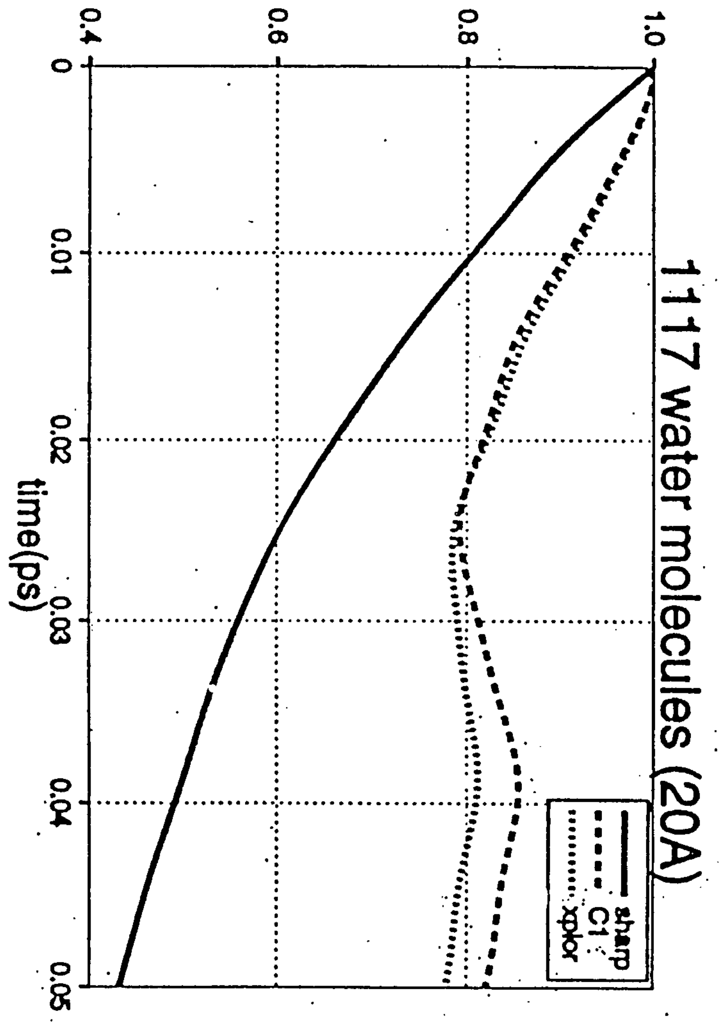


#2 use long timestep for smooth part
e.g. 20 fsecs

⟹ 50% premium for full electrostatics

6

<F(t) - F(t-T)> for Oxygen


1117 water molecules (20A)
0.4
0.8
0.8
1.0
0
0.01
0.02
0.03
0.04
0.05
time(ps)
sharp
C1
xplor

long-timestep methods

x positions, p momenta, M masses

$$\frac{d}{dt} x = M^{-1} p$$

F short-range forces, G smooth forces

$$\frac{d}{dt} p = F(x) + G(x)$$

idea
(i) sample G infrequently
(ii) solve "reduced problem" involving F and G-samples

worthwhile if either
(i) $\text{cost}(G) \gg \text{cost}(F)$ and/or
(ii) long timesteps w/ F are economical

example 1. *LIN*

example 2. impulse method
aka Verlet-I, RESPA

$$\bar{P}^n = P^n + \frac{\Delta t}{2} G(X^n);$$

$$(X^{n+1}, \bar{P}^{n+1}) = \Delta t\text{-flow of}$$

$$\frac{d}{dt} x = M^{-1} p, \quad \frac{d}{dt} p = F(x)$$

applied to $(X^n, \bar{P}^n)$;

$$P^{n+1} = \bar{P}^{n+1} + \frac{\Delta t}{2} G(X^{n+1})$$

symplectic, reversible
if F + G ...

e.g. 2-timestep method

$O(\Delta t)$ impulse

example 3. Verlet-X method

$$G^n = G(X^n);$$

$$(X^{n+1}, \bar{p}^{n+1}) = \Delta t\text{-flow of}$$

$$\frac{d}{dt}x = M^{-1}p,\ \frac{d}{dt}p = F(x) + G^n$$

applied to (X^n, p^n);

$$p^{n+1} = \bar{p}^{n+1} + \frac{\Delta t}{2}(G(X^{n+1}) - G^n)$$

$O(\Delta t^2)$ impulse

example 4.

Verlet-II method

$$G^n = G(X^n)$$

$(X^{n+1}, P^{n+1}) = \Delta t$-flow of

$$\frac{d}{dt}x = M^{-1}p, \quad \frac{d}{dt}p = F(x) + G^n + 3\frac{t^{n+\frac{2}{3}}-t}{\Delta t}(G^n-($$

applied to (X^n, P^n)

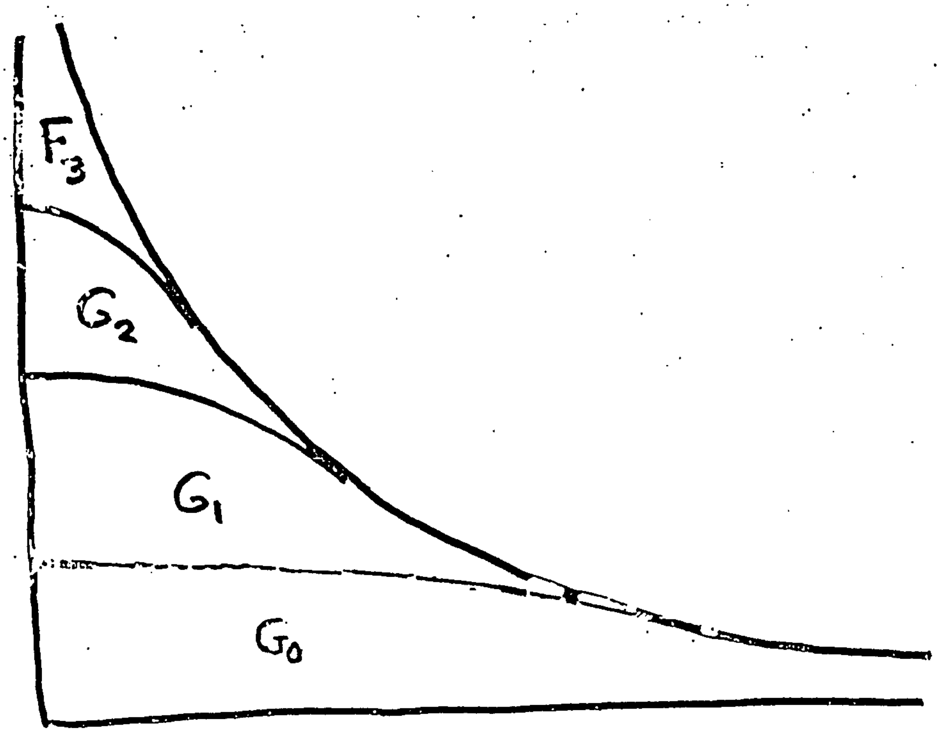
numerical experiments w/ water in progress μltiple timesteps $\frac{d}{dt}x = M^{-1}p$ $\frac{d}{dt}p = F_0(x)$

$$\begin{array}{rl} F_0 = & G_0 + F_1 \\ = & G_0 + G_1 + F_2 \\ \vdots & \\ = & G_0 + G_1 + \cdots + G_{n-1} + F_n \\ & \Delta t \quad 2^{-1}\Delta t \qquad 2^{1-n}\Delta t \quad 2^{-n}\Delta t \end{array}$$

j-th level timestep (impulse method)

```
step($2^{-j}\Delta t, F_j$) =
    if j < n then
        $p := p + 2^{-j-1}\Delta t\, G_j(x)$;
        step ($2^{-j-1}\Delta t, F_{j+1}$);
        step ($2^{-j-1}\Delta t, F_{j+1}$);
        $p := p + 2^{-j-1}\Delta t\, G_j(x)$
    else
        $p := p + 2^{-n-1}\Delta t\, F_n(x)$;
        $x := x + 2^{-n}\Delta t \cdot M^{-1}p$;
        $p := p + 2^{-n-1}\Delta t\, F_n(x)$
    end if
```

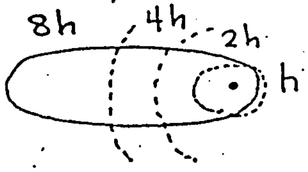
Flow of $\frac{d}{dt}x = M^{-1}p$ $\frac{d}{dt}p = 0$

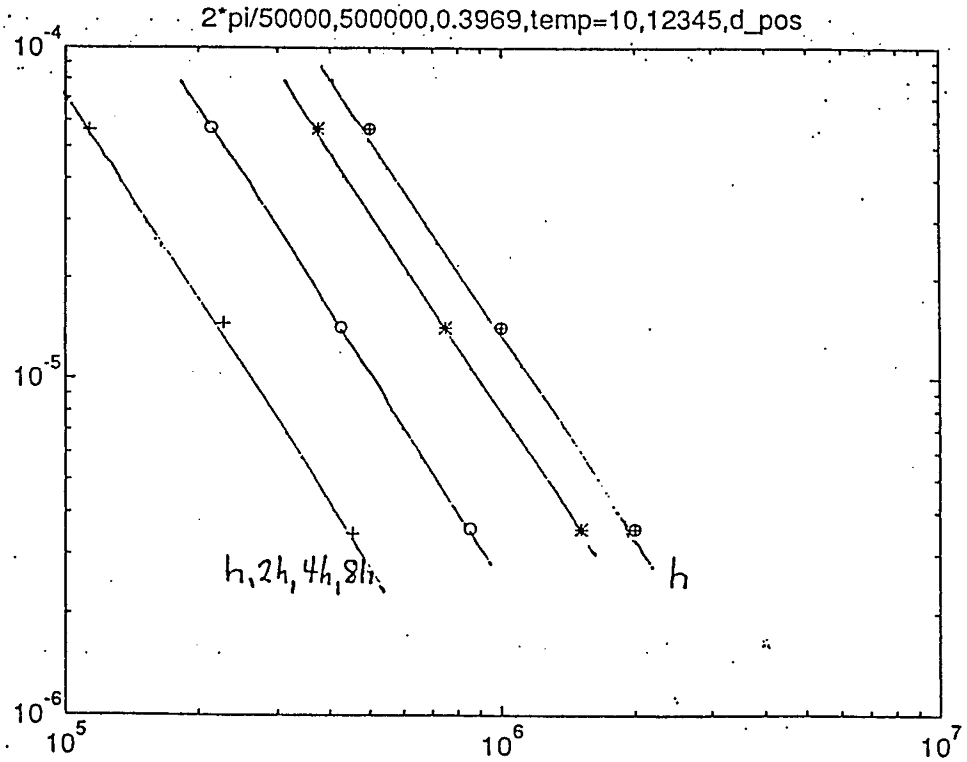

F_3
G_2
G_1
G_0

Z19A

first an example

Kepler problem, eccentricity 0.9





Reduced Variable Molecular Dynamics

Oren M. Becker, Tel-Aviv U.

Harvard U.

Moldyn Inc.

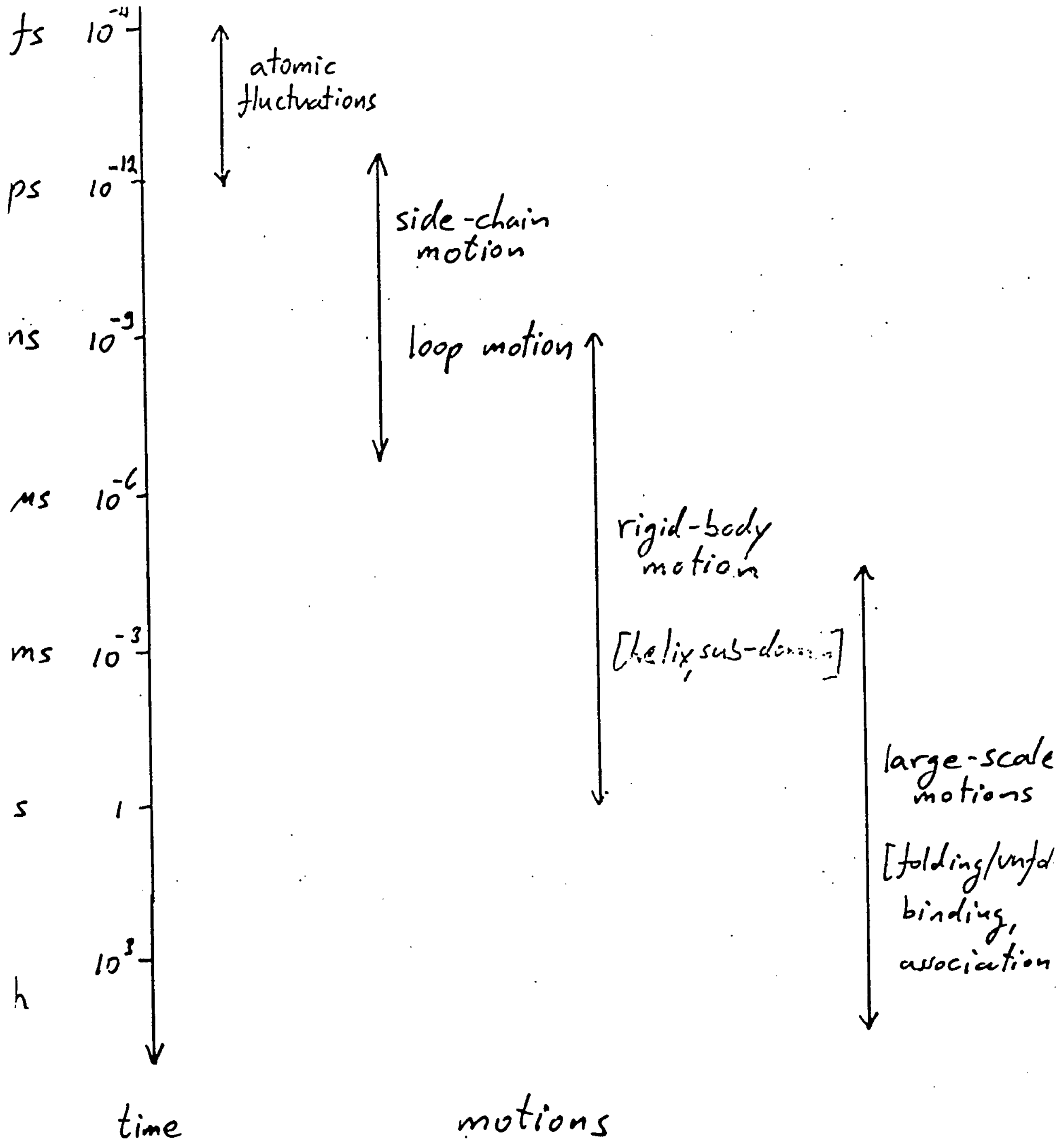
becker@sapphire.tau.ac.il

Proteins

Large system

Condensed

3D structure



Protein Dynamics

Goal:

Study Dynamics of Large Biomolecules on "biological" time scales.

⇒ (1) Thermodynamic properties.

(2) Biological function & activity.

(3) Design new drugs or proteins.

Molecular Dynamics

1. Atom-based models

2. Empirical force-field:

$$E = \sum \text{bonds} + \sum \text{angles} + \sum \text{dihedrals} + \sum \text{electrostatic} + \sum \text{vdW} + \ldots$$

3. Numerical integration of Newton's equation of motion:

$$m_i r_i = -\nabla_i V(r_1, \ldots, r_N)$$

in Cartesian coordinates

ie, $r_i = x_i \hat{n}_x + y_i \hat{n}_y + z_i \hat{n}_z$

Limited Length of Simulation

1. Small integration time-step

$$\tilde{\nu}_{max} \approx 3000\ cm^{-1} \quad \Rightarrow \tau_{min} \approx 10\ fs$$

$$\frac{\tau_{min}/2}{\Delta t} \gtrsim 10 \quad \Rightarrow \Delta t = 0.5\ fs$$

2. N^2 non-bonded interactions

   (>90% of computation time)

3. Treat <u>all</u> atoms on equal basis, but often one is interested only in a smaller sub-system (eg, active site).

Methods to Extend Simulation Time

1. Constraints to eliminate high frequency motions.

   If eliminating X-H bonds $\Rightarrow \bar{\nu}_{max} \approx 1500\ cm^{-1}$

   $\Rightarrow \Delta t = 1\ fs$

   e.g. SHAKE iterative $s_k = \frac{(r_{ij}^2 - d_{ij}^2)}{d_{ij}^2} < \varepsilon$

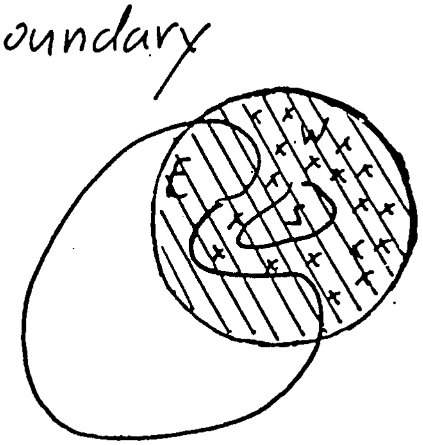
2. Cut off on non-bonded interactions

   (today → multipole expansion electrostatics).

3. Reduce no. of DOF (aproximations)

   (i) Vacuum simulations

   (ii) Stochastic boundary



Today simulations of a 1500 atom system (~ 100 a.a.) are typically taken to 1 ns.

Larger systems and/or longer times only with Approximate Methods:

(I) Stochastic Dynamics

propagate Langevine EOM:

$$m_i \ddot{r}_i = -\nabla_i V(r_1, \ldots, r_N) - \alpha_i v_i + R(t)$$

(II) Harmonic Dynamics

propagate along normal-modes. (very limited)

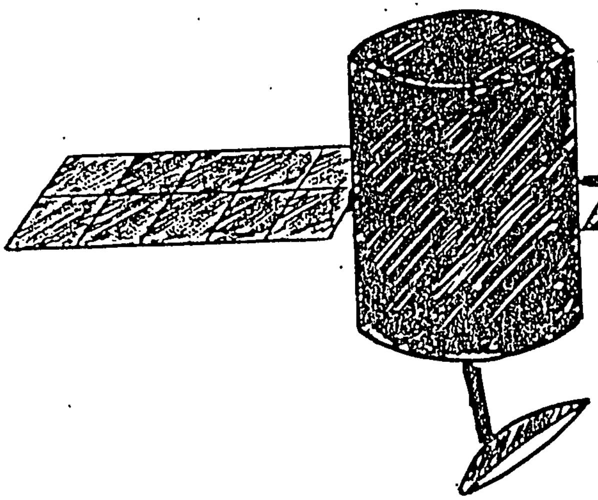
(III) Rigid body simulations

Get to longer times by focusing on biologically interesting motions, while retaining a correct but cheep description of faster motions.

Variable Reduction Techniques

Generate EOM of mechanical systems with rigid and flexible components.

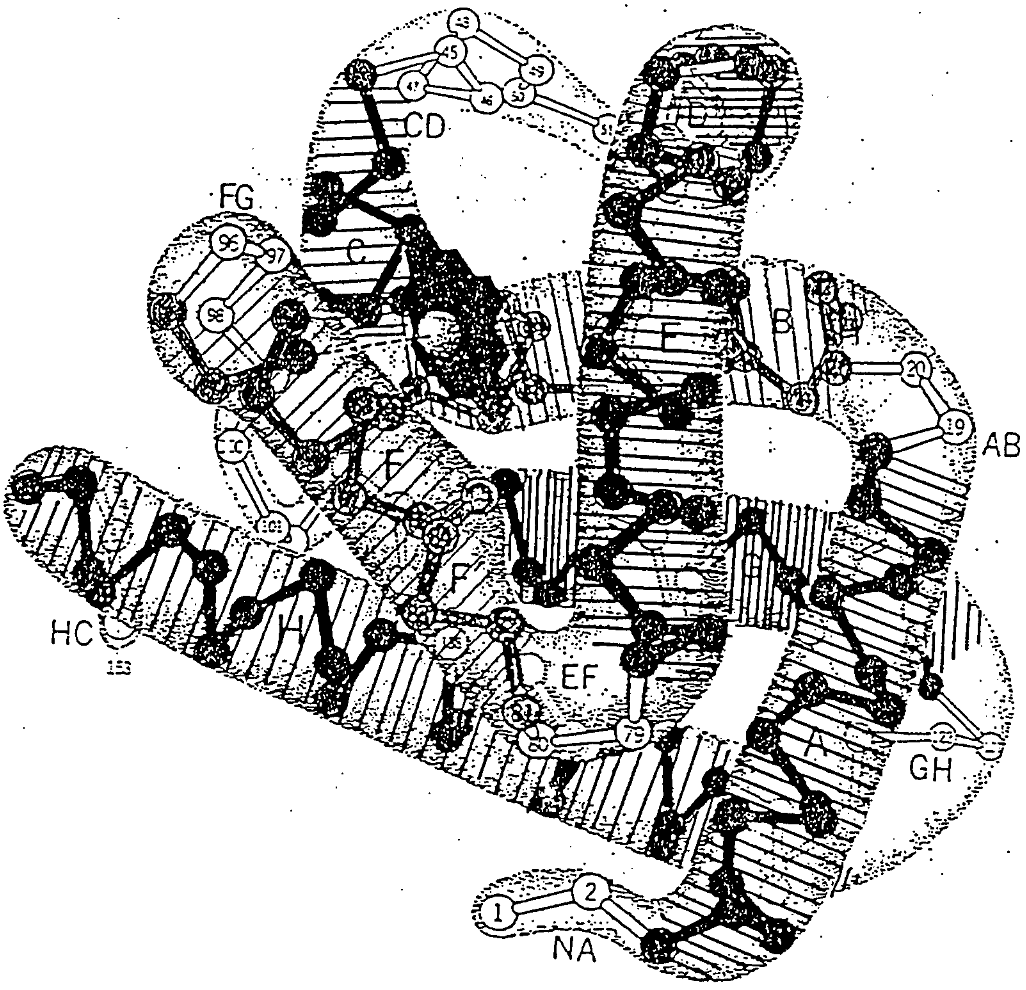
Aerospace dynamics



Bodies connected by hinges (constraints).

rigid bodies - Euler-Newton EOM (rotation & translation)

flexibility - Low frequency normal-modes.



Deoxy Myoglobin

Reduced Variable Molecular Dynamics

1. Substructuring

Group atoms together to define 'Bodies'.

Criterion: Relative displacements small
compared to inter-body
displacements.

e.g. α-helix

but not loop-regions, terminal arms.

$\Rightarrow$ (*) Mixture of 'bodies' and 'particles'.

(*) All bodies must be flexible.

2. Equations of Motion

Particles – Newton EOM (r)

Bodies – Euler-Newton EOM (r, θ)

Hinge constraints

3. Flexibility

Modeled by a reduced set of "component" normal-modes (low frequency). remove high freq., i.e., local vibrations.

For each body generate modes in the field of the rest of the molecule.

'Body-modes' are coupled to large-scale motion through rigid-body kinematics.

Separation of variables:

$$U(R;t) = \sum_{k=1}^{N_m} \underset{\text{eigenvectors}}{\Phi_k}(R)\, \underset{\text{time dependent amplitude}}{\xi_k}(t)$$

Component mode synthesis:

$$m_A \ddot{X}_A = f_{AA}(X_A) + f_{AB}(X_A, X_B)$$

solve for $X_A = \Phi(R)\xi(t)$ to get

$$(\Phi^T m_A \Phi)\ddot{\xi}_A = \underset{\text{linear}}{-(\Phi^T K_{AA} \Phi)\xi_A} + \underset{\text{non-linear}}{\Phi^T f_{AB}(\xi_A, \xi_B)}$$

Application

First tests AMBER-MBO(N)D

J.D. Turner et al. 1993

Extensive interface CHARMM-MBO(N)D

O. M. Becker et. al.

Expected speed-up

(I) Large reduction in nº of DOF.

(II) Simplify energy evaluation.

(III) Larger integration time-step. (MTS)

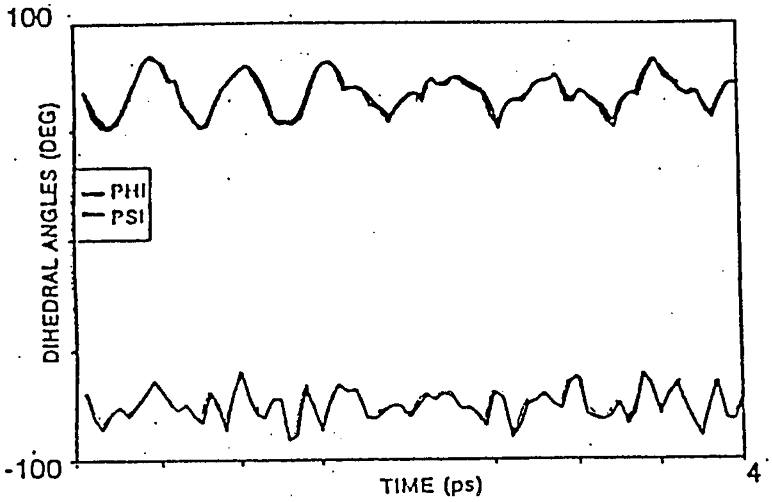
(IV) Eficient constraint algorithm.

expected 10-100-fold speed-up.

Glycine dipeptide

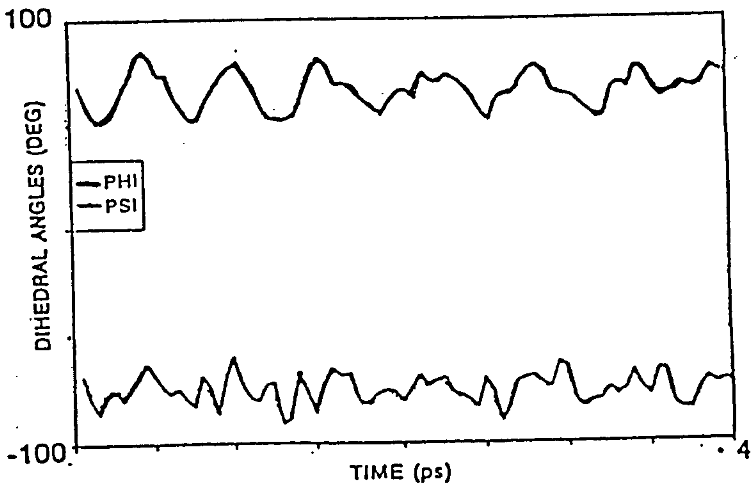
SHAKE

Verlet, TIME =4ps, ΔT = 0.5fs



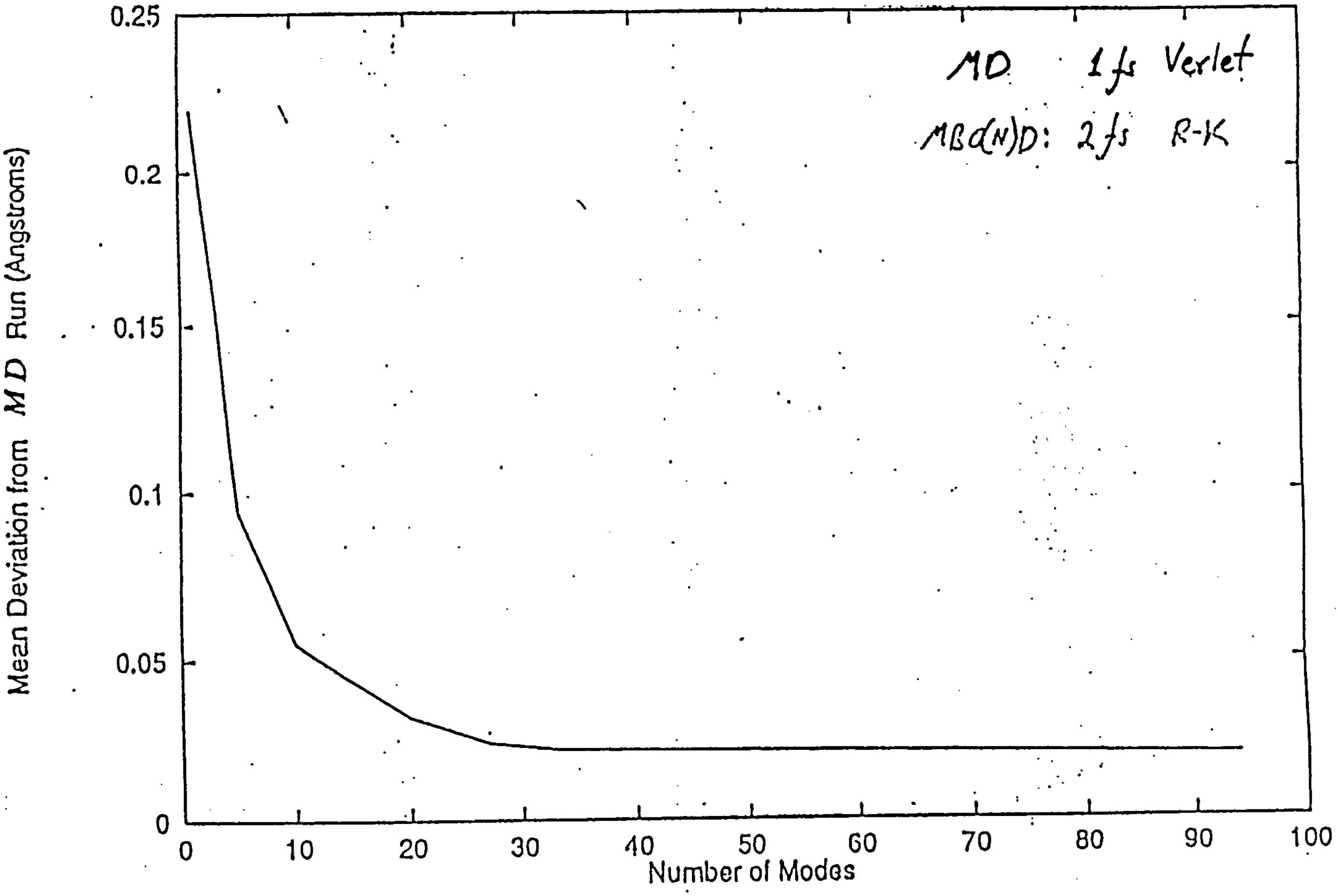
MBO(N)D

RK4, TIME = 4ps, ΔT = 1fs

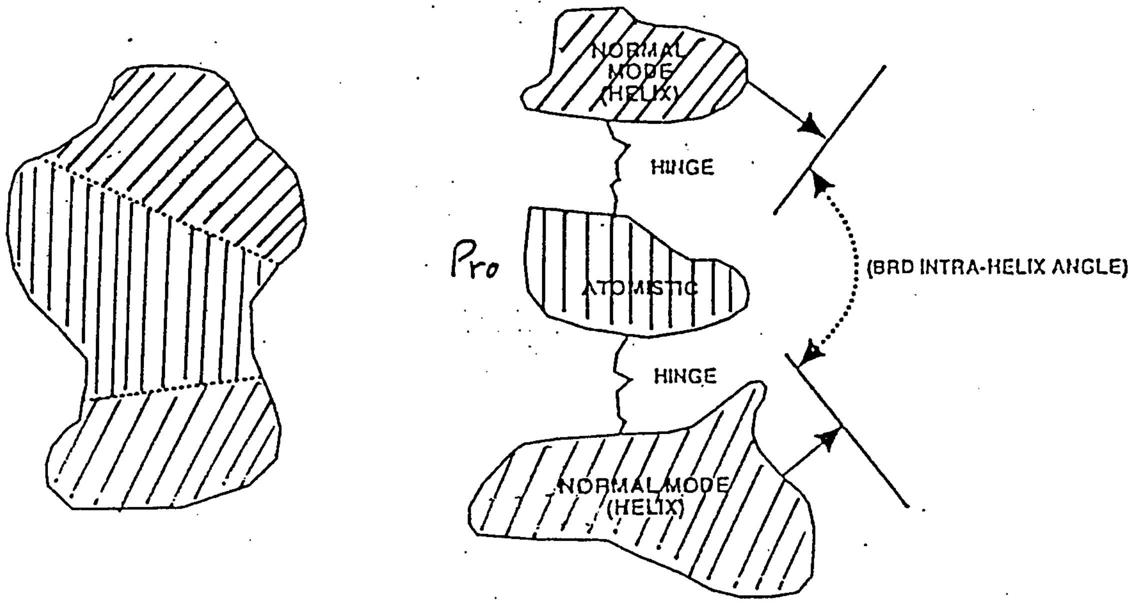


Deca-Gly, 20 ps

α-helix, 13 a.a.

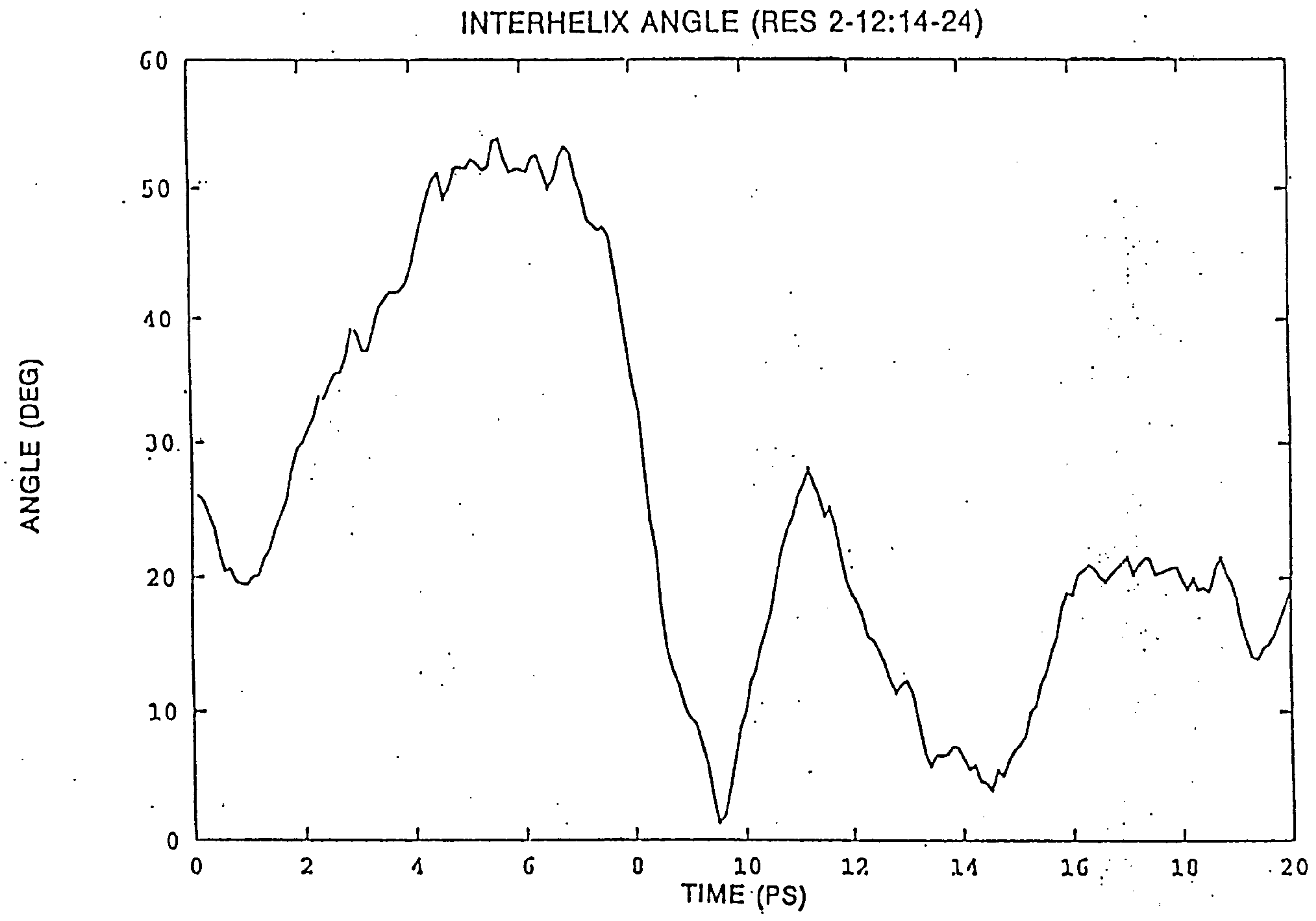


25 aa from Bacteriorhodopsin (237 atoms)



GLOBAL BEHAVIOR

SUBSTRUCTURE MODELS


INTERHELIX ANGLE (RES 2-12:14-24)
ANGLE (DEG)
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
TIME (PS)

25-aa. from Bacteriorhodopsin

Type of run	R.m.s. fluctuation (degrees)
MD	17.55
One flex body	8.05
Three rigid	1.5
Flex (8) flex (4) flex (8)	10.79
Flex (2) particle flex (2)	12.01
Flex (8) particle flex (8)	18.18

Summary

Reduced Variable MD – new possibility for long simulations of large biomolecules.

Open Questions

- partitioning strategies.
- mode generation & update.
- accuracy vs. efficiency.

Present applications

4-helical bundle (rop)

8-helical bundle (myoglobin)

Advanced applications

(1) Enzyme hinge-bending motion
e.g. Liver Alcohol Dehydrogenase (2x3000 heavy atoms)

(2) Membrane dynamics.

Oren M. Becker	(TAU)
Leo S.D. Caves	(York)
Robert Nagle	(Harvard)
Herman Vlijman	(Harvard)
Martin Karplus	(Harvard)
Hon M. Chun	(Moldyn inc.)

Algebraic Multigrid (AMG)

Multi-level strategy for solving matrix problems $A\,U = f$

Klaus Stüben

GMD/SCAI
Schloß Birlinghoven
D-53757 St. Augustin
Germany

A *sparse* and (approx.) *positive type:*

$a_{ij} \le 0 \ (i \ne j)$ and $\sum_j a_{ij} \ge 0$

References on algebraic multigrid:

Brandt, A.: *Algebraic multigrid theory: the symmetric case,* Appl. Math. Comp. 19 (1986)

Ruge, J.; Stüben, K.: *Algebraic Multigrid.* In: "Multigrid Methods", McCormick, S. (ed.), Frontiers in Applied Mathematics, Vol. 5, SIAM, Philadelphia (1987)

Algebraic characterization of *smoothness*

GS-relaxation for i-th unknown: $u_i \rightarrow \bar{u}_i$ where

$$\boxed{\bar{u}_i = \frac{1}{a_{ii}}\Big(f_i - \sum_{j\neq i} a_{ij} u_j\Big) = u_i + \frac{d_i}{a_{ii}}}$$

$d_i := f_i - \sum_{j} a_{ij} u_j$ = residual *before* relaxing

$$\longrightarrow \quad \bar{e}_i = e_i - \frac{d_i}{a_{ii}}$$

The error e is *smooth* $\Longleftrightarrow$ relaxation is stalling

$\Longleftrightarrow$ $\bar{e}_i \approx e_i$ *for all i*

$\Longleftrightarrow$ $|d_i| \ll a_{ii}|e_i|$ *for all i*

i.e., the residual is much smaller than the error!

Since $d_i = a_{ii} e_i + \sum_{j\neq i} a_{ij} e_j$

smooth error e satisfies approx.

$$a_{ii} e_i = -\sum_{j\neq i} a_{ij} e_j \quad \textit{for all i}$$

This relation is the basis both for coarsening and interpolating

<u>Remember (variational principle!):</u>

Smooth error has to be "close to" $\mathcal{R}$:= range(interpol)

Note that, generally, (algebraically) smooth error is NOT necessarily (geometrically) smooth and vice versa!

General approach

Fix GS as smoothing method

Construct subset of unknowns ("coarser levels")
and "interpolation" such that *

- smooth error is interpolated well, i.e.
  - smooth error "close to" $\mathcal{R} := \text{range}(I_H^h)$
- coarsening is "efficient", i.e.
  - fast reduction of # unknowns
  - no serious fill-in on coarser levels

Two-level method by variational approach

Multilevel method by recursive application

*) This construction is part of an AMG algorithm!

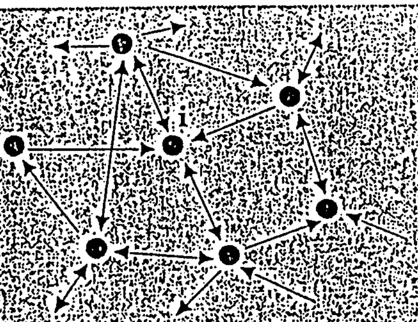
Graph representation of matrix A

With each unknown, i, associate a "point":

$\Omega := \{i\}$ = *fine "grid"*

$N_i := \{j \neq i : a_{ij} \neq 0\}$

neighborhood of i



$\overset{i}{\bullet} \longrightarrow \overset{j}{\bullet}$ *i is connected to j:* $a_{ij} \neq 0$

$\bullet \longleftarrow \bullet$ *j is connected to i:* $a_{ji} \neq 0$

Requested: splitting $\Omega = F \cup C$
(C-points: coarser "grid")

How to interpolate *smooth* error e
at F-points from those at C-points?

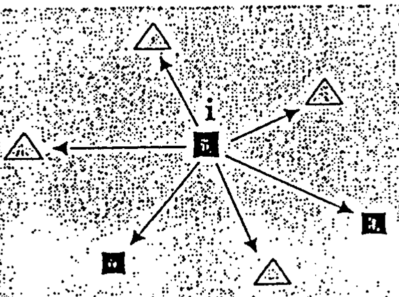
■ : F-points △ : C-points

$F_i := F \cap N_i \quad C_i := C \cap N_i$

smooth error satisfies:

$$\boxed{a_{ii} e_i} = -\sum_{j \in C_i} a_{ij} e_j - \underbrace{\boxed{\sum_{j \in F_i} a_{ij} e_j}}_{\text{has to be "eliminated"}}$$

?



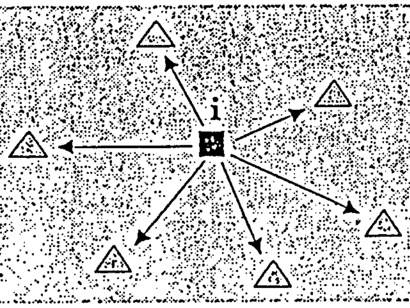
A simple but inefficient coarsening strategy

Choose C such that for all $i \in F$: $N_i \subset C$

i.e., F-points are *totally decoupled!*

$i \in F$: $F_i = \emptyset$

$$\boxed{a_{ii} e_i} = -\sum_{j \in C_i} a_{ij} e_j$$



Two-level method:

GS-relaxation with C-F-ordering

Above error relation holds *exactly* at F-points!

Coarse-grid correction

$$\Omega_h := \Omega, \quad \Omega_H := C$$

$$(I_H^h e^H)_i := \begin{cases} e_i^H & (i \in C) \\ -\sum_{j \in C_i} a_{ij} e_j^H / a_{ii} & (i \in F) \end{cases}$$

Is a *direct solver*, i.e. residual=0 after one cycle!
(After relaxation of F-points: $e^h \in \mathcal{R} := \mathrm{range}(I_H^h)$)

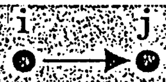
However, recursive application extremely inefficient:
Drastic fill-in + slow coarsening!

A practical coarsening strategy

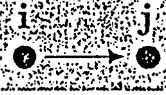
Idea: distinguish *strong* and *weak* connections:

i is *strongly* connected to j if

$$-a_{ij} \ge \alpha \max_{k \ne i} \{-a_{ik}\} \qquad (0 < \alpha \le 1)$$



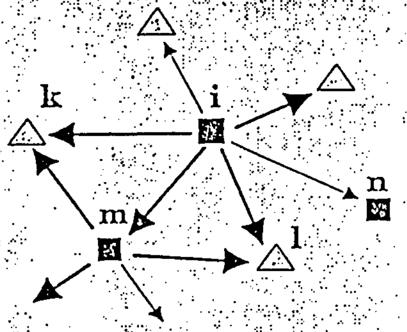
therwise, we call it *weakly* connected.



Typical value in practice is $\alpha = 0.25$

Representative situation:

$$a_{ii} e_i = -\sum_{j \in C_i} a_{ij} e_j - \sum_{j \in F_i} a_{ij} e_j$$



"Elimination" of n and m:

$e_n \rightarrow e_i$ (weak connection!)

$$e_m \rightarrow \frac{a_{mk}\, e_k + a_{ml}\, e_l}{a_{mk} + a_{ml}}$$

General requirements:

) For all $i \in F$: all *strong* neighbors should be

-- either C-points used in interpolation (like k,l)

-- or strongly connected to such points (like m)

) The # of C-points should be *as small as possible*

hese are the basic objectives which allow for an automatic and purely algebraic coarsening process!

(1) is responsible for good interpolation of smooth error
(2) is important for low-cost cycles

MG convergence (unit square, uniform grid)

problem		AMG time/cycle setup	AMG ρ	BOXMG* time/cycle setup	BOXMG* ρ
$-u_{xx}-u_{yy}$		0.30 1.90	0.056	0.33 0.23	0.015
$-\varepsilon u_{xx}-u_{yy}$	0.001	0.33 1.53	0.084	0.33 0.24	0.043
	0.01	0.34 1.75	0.093	0.33 0.24	0.107
	0.1	0.38 2.70	0.058	0.33 0.24	0.088
	0.5	0.31 1.90	0.069	0.32 0.24	0.036
	2	0.31 1.91	0.079	0.32 0.23	0.037
	10	0.38 2.65	0.087	0.32 0.24	0.084
	100	0.33 1.74	0.093	0.32 0.24	0.104
	1000	0.32 1.55	0.083	0.32 0.24	0.043
$(100^{x+y-1}u_x)_x-u_{yy}$		0.33 1.95	0.080	0.32 0.24	0.044
$-\nabla(a\nabla u)$ **		0.32 1.98	0.069	0.32 0.24	0.044

$$** \quad a := \begin{array}{|c|c|}\hline 10 & 100 \\ \hline 1 & 1000 \\ \hline\end{array}$$

* BOXMG (J. Dendy) was used in its most robust form (i.e. alternating line relaxation, Galerkin,...)

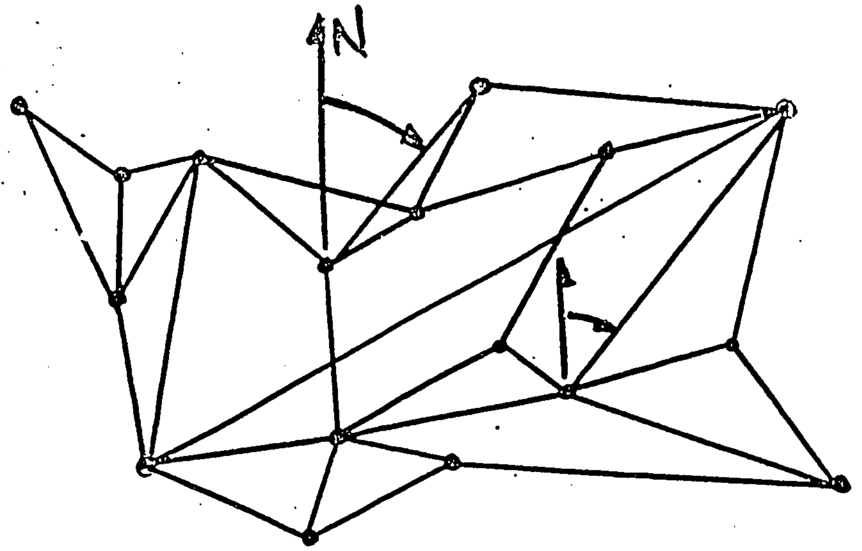
AMG Applied to Functional Minimization Problems

(Unstructured meshes, vector unknowns)

John Ruge
Achi Brandt
S. McCormick
Zhou Xingjun
etc.

Supported by AFOSR, Veritas Research,...

Geodetic Survey Problems



- Sites with unknown locations (x_i, y_i)

— Measurements (distance) d_{ij}

Varying accuracy (different methods)

Measurements (angles) θ_{ij}

= Find (x_i, y_i) that give "best" fit to data

$$\sum_{ij} w_{ij} \left(|(x_i,y_i)-(x_j,y_j)| - d_{ij} \right)^2 + \sum_{ij} \alpha_{ij} \left(\tilde{\theta}_{ij} - \theta_{ij} \right)^2$$

computed angle

Truss (Beam) Structures

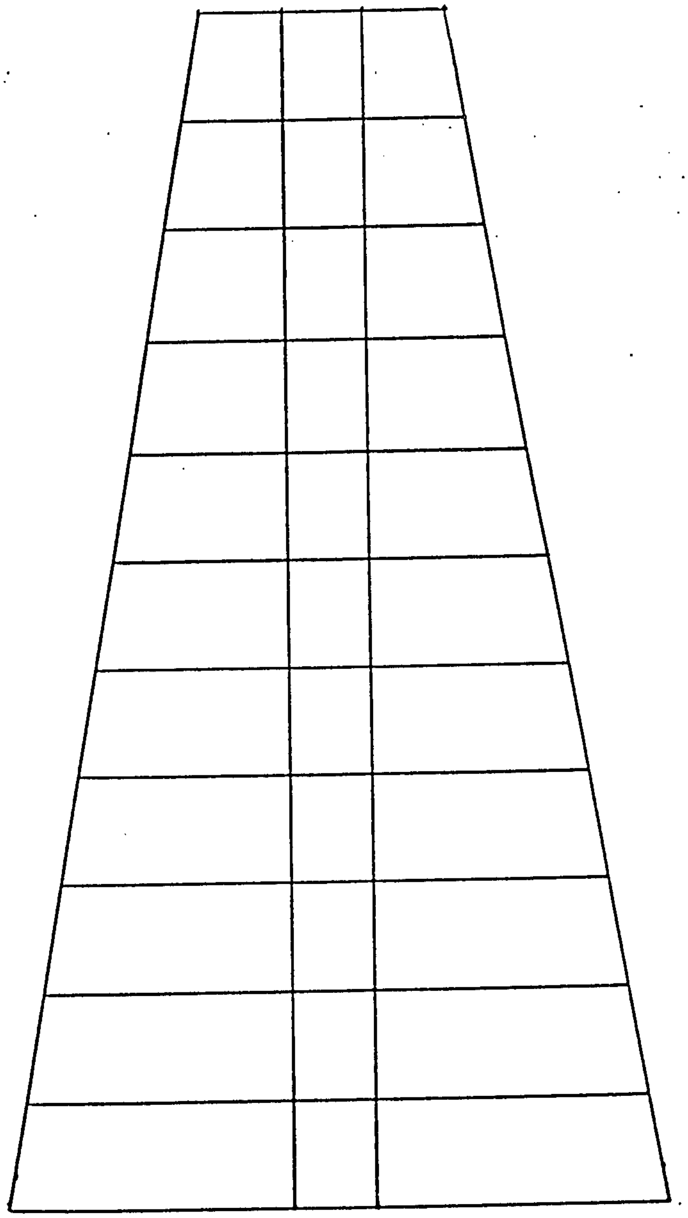
- Small displacements -
  beams behave like springs
  (force = $k \Delta l$)

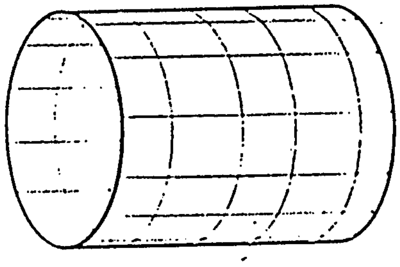
- Joints <u>hinged</u> or stiff. (gives angle term

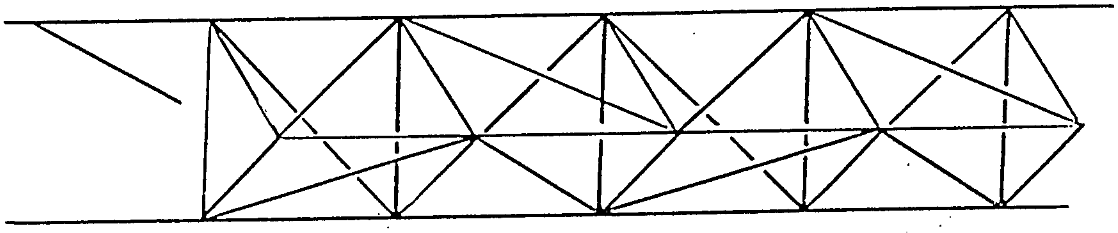
- Formulate energy functional

  $$\sum a_{ij} (\Delta l_{ij})^2$$

(Or sum forces, set = 0)



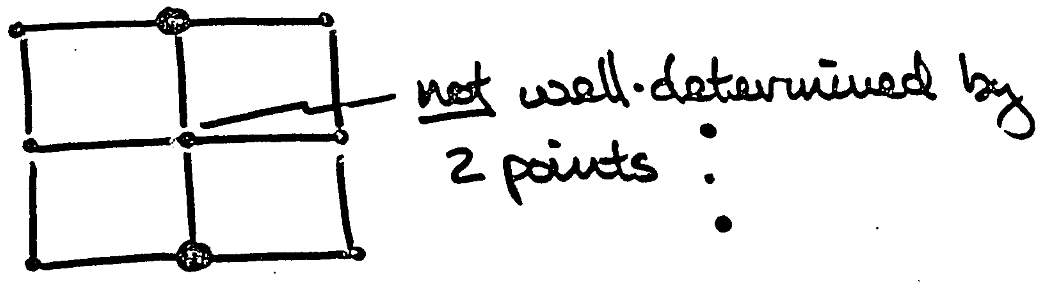




Vector unknowns & AMG

- Coarsen "pointwise"
- "Strong connection" now <u>vector</u>

⇒ each component $(x_i, y_i, \ldots)$ must be well-determined by C.



∴ Requires modification of AMG coarsening process.

For now, <u>ignore</u> CG choice.

Assume functional of the form:

$$\sum_{ij} \alpha_{ij} \left(\beta_{ij}(x_i - x_j) + \gamma_{ij}(y_i - y_j) - \delta_{ij} \right)^2$$

(i.e. linearized)

Minimization gives the problem

$$Au = f$$

$$u = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \end{bmatrix}$$

$A = \sum A_{ij}$ $\quad$ A_{ij} from ij term.

More generally, denote terms as A

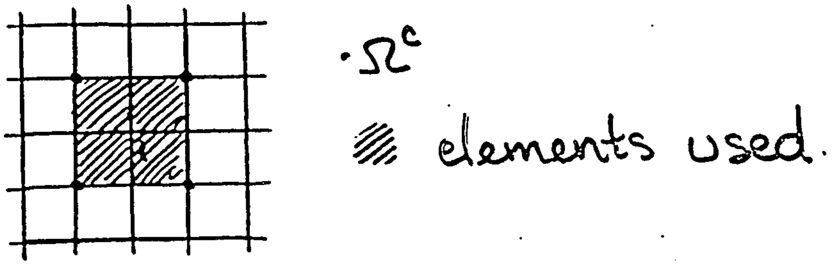
"Element" interpolation

N_i - neighboring points of i.

E_i - elements with i as a node

Instead of minimizing $v^TAv = v^T\sum_{\alpha} A_{\alpha} v$

minimize only $v^T\sum_{\alpha\in E_i} A_{\alpha} v$ to get v_i.



- point i interpolates only from points in $N_i \cap \Omega^c$.

To define element interpolation to i:

Consider the following 3 sets

$G = \{i\} \cup \{j \notin \Omega^c : j \in N_i \ \& \ E_j = E_i\}$

$F = \{j \in N_i ; \ j \notin G, \ j \notin \Omega^c\}$

$C = \{j \in N_i : \ j \in \Omega^c\}$

Renumber the points: G F C · ——

$$\sum_{\alpha \in E_i} A_\alpha = \begin{bmatrix} A_{GG} & A_{GF} & A_{GC} & \\ A_{FG} & A_{FF} & A_{FC} & 0 \\ A_{CG} & A_{CF} & A_{CC} & \\ & 0 & & 0 \end{bmatrix}$$

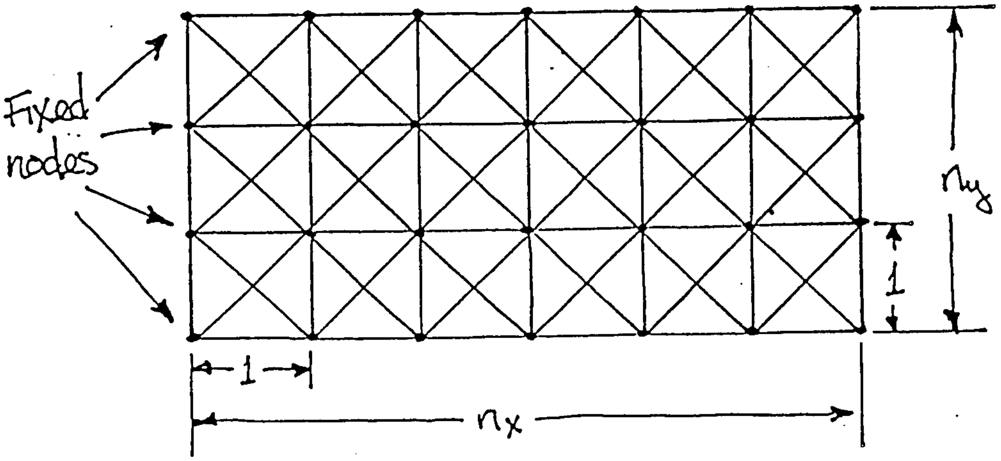
Partition $v = (v_G, v_F, v_C, v_R)^T$

Then, with v_C fixed and $v^T \sum_{\alpha \in E_i} A_\alpha v$ minimal,

$$\underline{v_G = -(A_{GG} - A_{GF} A_{FF}^{-1} A_{FG})^{-1} (A_{GC} - A_{GF} A_{FF}^{-1} A_{FC}) v_C}$$

2-D Beam Structures

- Hinged joints
- Beams subject to compression/extension
- Equations derived from Hooke's Law.



		V-cycle Rates	
n_x	n_y	element int.	linear int.
16	16	.26	.20
32	8	.27	.64
64	4	.28	.97
128	2	.28	.93 +

A. Brandt & D. Bai

Multiscale Methods in Molecular Dynamics

MULTI-SCALE (MS) RESEARCH METHODOLOGY

Problem size (# atoms) = n

Objective: Total work = $O(n)$

Research stages:

increasingly complicated problem

But at each stage $n \rightarrow \infty$

At each stage add one complication, insisting on still obtaining $O(n)$

DIMENSIONS

1D, 2D, 3D

TOPOLOGIES

Many non-bonded
- Identical atoms
- Identical small molecules
- Several species

Bonded chains :
- Identical atoms, stretched
- Helics
- Identical amino acids

Chains + solvent

General

OBJECTIVES	MS
Energy minimization	
• Near-minimum start	MG, AMG
• Far start	ms annealing
• Homogenization	
Equilibrium statistics	
• O(n) per sample	ms
• O(1) per sample	Monte-Carlo
⇓	
• Homogenization	
Dynamics	
• Large Δt	MG, AMG
• Very large Δt	ms annealing

Stochastic dynamics

ms Monte-Carlo at each time step

POTENTIALS

Bonds

- Harmonic (1) constant coupling
  (2) strongly variable
- Bond length
- Length + angle
- Length + angle + torsion

Non-bond, local

- Van der waals
- Hydrogen bonding

Non-local

- Electrostatics : (1) constant dielectric
  (2) variable dielectric

Combinations

Stochastic Dynamics

At time t^n: position vector X^n
velocity vector V^n
potential energy $E(X^n)$.

Deterministic implicit time step:

$$-\nabla E(X^{n+1}) = p^{n+1}$$

$$p_k^{n+1} = m_k \frac{\delta V_k}{\delta t}, \; \delta V_k = V_k^{n+1} - V_k^n, \quad V^{n+1} = \frac{X^{n+1} - X^n}{\delta t}$$

$\Longleftrightarrow$ X^{n+1} minimizes $H_n(X^{n+1})$

$$H_n(X^{n+1}) = \frac{1}{2} \sum m_k (\delta V_k)^2 + E(X^{n+1})$$

Stochastic time step:

$$P(X^{n+1}) \sim e^{-\beta H_n (X^{n+1})}$$

$$\beta = \frac{1}{k_B T}$$

Results of the stochastic step :

- $< p^{n+1} > = -\nabla E(x^{n+1})$
- $<(p_k^{n+1})^2> - <p_k^n>^2 = m_k/\beta$
- Local components, with
  oscillation period $\ll \delta t$
  are fully thermalized:
  $$P(x^{n+1}) \sim e^{-E(x^{n+1})/k_B T}$$
- Large scale components, with
  oscillation period $\gg \delta t$
  satisfy Newton law.
- Easy multiscale equilibration

Relaxation

- Atom by atom $\left\{\begin{array}{l}\text{minimization}\\ \text{Monte-Carlo}\end{array}\right.$

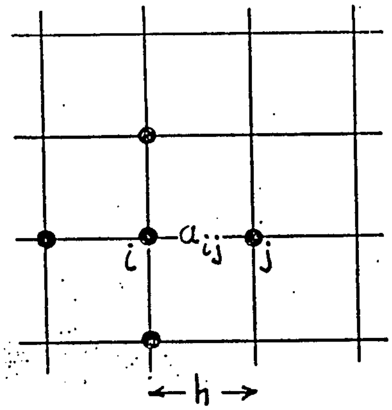
- Only local forces

  including local part of global forces

- For each atom – use its natural (internal) coordinates

$$E(u) = \frac{1}{2} \sum_{<i,j>} a_{ij} \left(\frac{u_i - u_j}{h} \right)^2 .$$

$$- \sum_i f_i u_i$$



$$E(U) = \min_u E(u)$$

$$0 = \left. \frac{\partial E}{\partial u_i} \right|_{u=U} = \frac{1}{h^2} \sum_{<j,i>} a_{ij} (u_i - u_j) - f_i$$

$$\frac{1}{h^2} \sum_{<j,i>} a_{ij} (U_i - U_j) = f_i$$

$$\longrightarrow \frac{\partial}{\partial x} \left(a \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(a \frac{\partial U}{\partial y} \right) = f$$

$$\Longleftrightarrow E(U) = \min E(u)$$

$$E(u) = \iint \left[\frac{a}{2} \left(u_x^2 + u_y^2 \right) - f u \right] dx\,dy$$

Algebraic Relaxation Theory

$Ax = b$ $\quad A = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} a_{11} \cdots a_{1n} \\ \vdots \\ a_{m1} \cdots \underset{mn}{a_{\cancel{mm}}} \end{pmatrix}$. $\quad$ error $e = x - \tilde{x} = \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}$

residuals $r = Ae = b - A\tilde{x}$

Normalized residuals $\bar{r}_i = r_i / |a_i| = a_i e / |a_i|$

Kaczmarz Relaxation: GS for $A\underbrace{A^T y}_{x} = b$

i.e. $\tilde{x} \leftarrow \tilde{x} - a_i^T (r_i / |a_i|^2)$, $(i = 1, \ldots, m)$.

Theorem. slow convergence $\Leftrightarrow |\bar{r}| \ll |e|$

$|e|^2$ is reduced per sweep at least by $|\bar{r}|^2 / \left(\max_i \frac{\sum_j |a_i a_j^T|}{|a_i a_i^T|} \right)^2$

∴ Slowly converging errors are special
They can be approximated by a smaller system (coarser grid). i.e., by $I_H^h v^H$

Discrete h-elliptic eqs. slow convergence

$\Leftrightarrow |L^h e^h| \ll |L^h| \, |e^h| \Leftrightarrow$ e^h smooth on scale h

A. Simple Energy Basins

real u:

Slow to equilibrate and be sampled : $I_H^h v^H$

Coarse Monte-Carlo Hamiltonian: $E^h(\tilde{u}^h + I_H^h v^H)$

$$\equiv E^H(v^H)$$

Multigrid cycle :

on each level :

1. Monte-Carlo passes
2. γ cycles on coarser
3. Monte-Carlo passes

Near equilibration and decorrelation in one cycle ($\gamma = 2$) .

Small work on coarser grids $(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots)$

Coarse-level variables

- Gridpoints
- Subset of atoms.

Interpolation.

- of displacements! (smooth)
- Reflects lowest eigen-modes
  - much more efficient than direct representation of all these modes
- Strong couplings are not constant
- Non linear relations retained (to the extent needed)

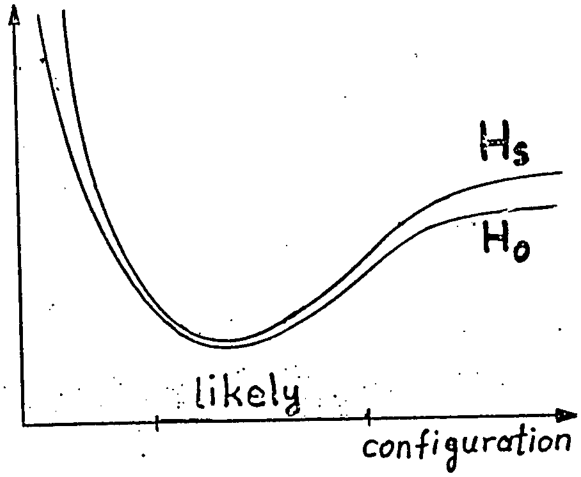
- Interpolation order
- Derivation of interpolation

electrostatics

Non linearity retained:

Smooth part of the forces (and residuals of the local part) participate (and change) in the coarse-level dynamics

STOCHASTIC HAMILTONIAN SIMPLIFICATION



ℓ_0 = Original Hamiltonian
obtained from fine grid $\Longrightarrow$ complicated

ℓ_s= Simplified Hamiltonian
FAS: original form + polinomial terms

$$\mathcal{H}_s \geqslant \mathcal{H}_0 \,, \quad \mathcal{H}_s \approx \mathcal{H}_0$$

Detailed Balance:

$$P(H_0 \rightarrow H_s) = e^{H_0 - H_s}\Big|_{\text{current}}$$

$$P(\text{freeze } H_s - H_0) = 1 - P(H_0 \rightarrow H_s)$$

Simpler: Approximate detailed balance
Aposteriori acceptance check

GOAL:

Fast minimization of energy functionals

MOTIVATION:

MD implicit schemes with large timestep

METHODOLOGY:

Minimize potential energy functionals in 1D, 2D, 3D.
Realistic couplings.

MULTI-SCALING CONSIDERATIONS

RELAXATION:

Simple, point-by-point

COARSENING:

AMG theory: Strongest coupling to coarse level

1:2 RATIO

INTEROLATION:

Precise for infimite coupling

RECURSIVENESS:

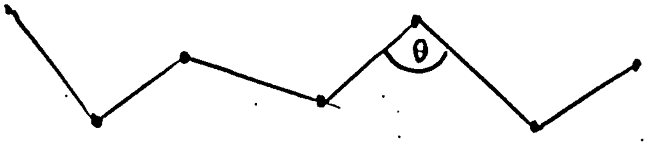
All levels

Investigated cases

(1) 1D chain

Nearest neighbor coupling

(2) 2D chain



bond + angle interaction

angle - 2nd neighbor

(3) 3D chains

bond + angle + torsion

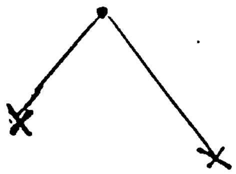
torsion - 3rd neighbor

PROPERTIES OF COARSE-LEVEL REPRESENTATION

Coarse atoms: more freedom

Bond constraints:

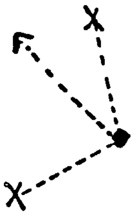
2D



x - coarse points

• - fine-only point

3D



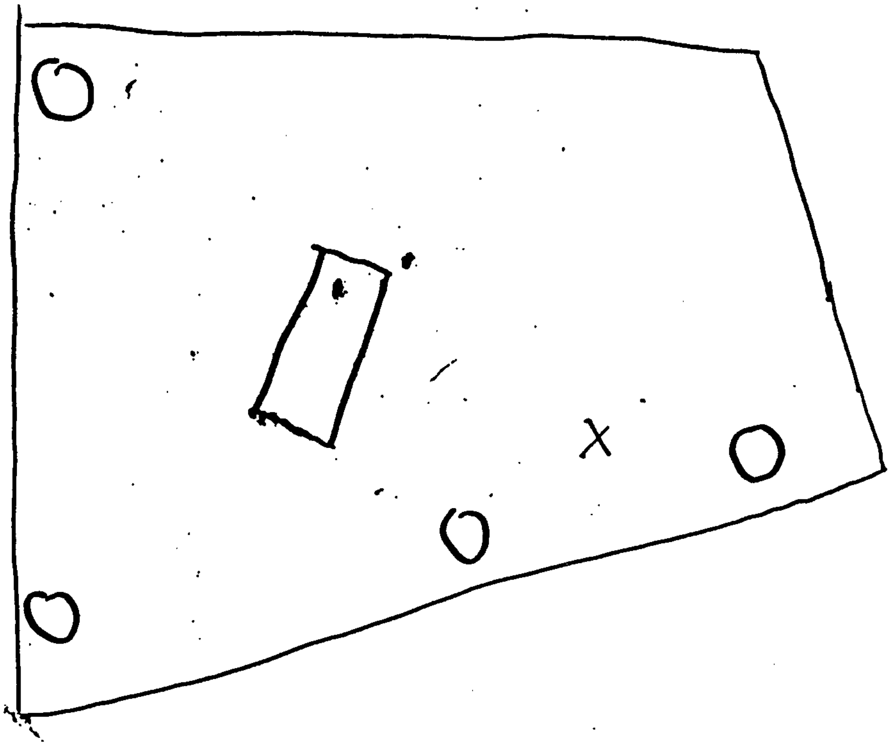
↗ - External field perpendicular to plane of atoms

NON-LINEARITIES

Typical reduction of strength of interaction on a coarse level in 2D.

	K_2									
	0.01		0.1		0.5		1.0		10.0	
	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$	$\langle K_x \rangle$	$\langle K_y \rangle$
Fine	1.04	3.0	1.4	3.0	3.0	3.0	5.0	3.0	40.0	3.0
coarse	0.05	0.09	0.5	0.9	2.5	4.5	5.0	9.0	49.0	90.0

K_x, K_y - Second energy derivatives in x, y directions



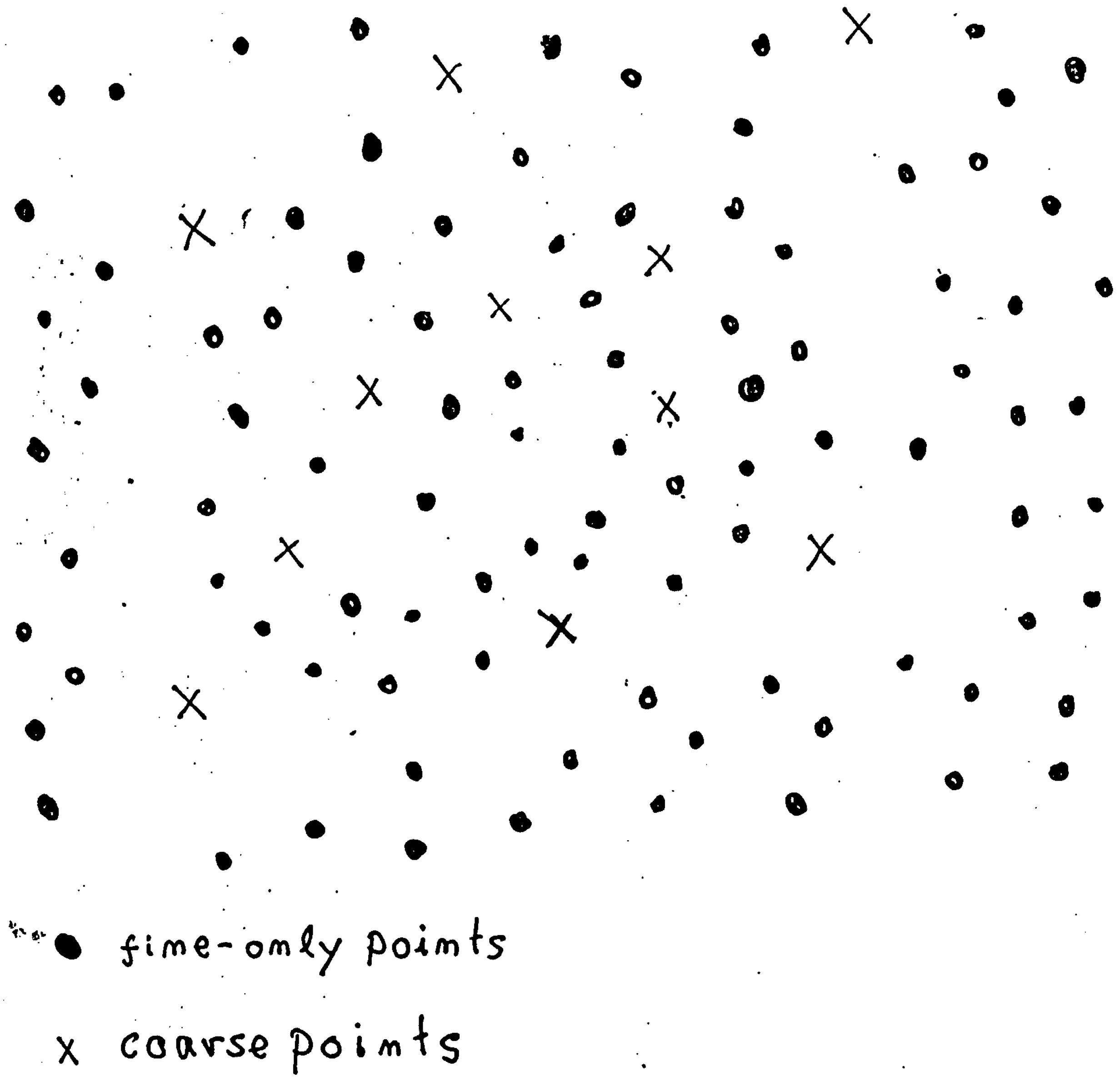
$\boxed{\cdot\ \ \cdot}$ - center points

⊗ - remote points

$$E_{\square} = (x_f^T\ x_c^T)\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}\begin{pmatrix} x_f \\ x_c \end{pmatrix} = x_f^T A x^f + 2x_f^T B x_c + x_c^T C x_c$$

minimum energy for a given x_c : $x_f = -A^{-1} B x_c$

Apply only to $\boxed{\cdot\ \ \cdot}$


● fine-only points
x coarse points

SLOWNESS OF RELAXATION

Slowed down by:

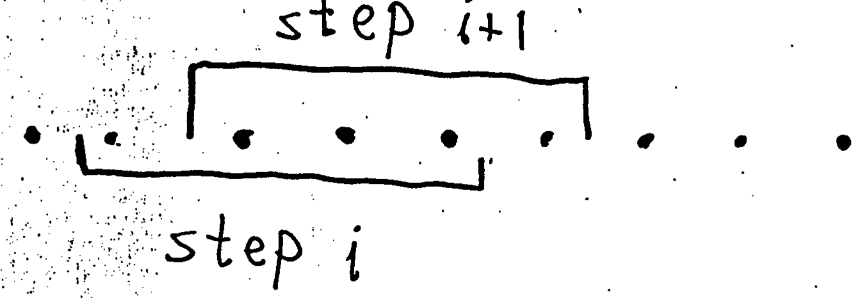
N - number of atoms

q - coupling ratio

$1-\varepsilon$: convergence rate

$\varepsilon < O(1/qN^2)$

simultaneous changes:



Effect of BS on convergence rates

3D generic chain

Energy terms:

$K_1(r-r_0)^2$ - bonding

$K_2(\theta-\theta_0)^2$ - angle interaction

$K_3(\varphi-\varphi_0)^2$ - torsion

$N=26$

$K_1 = 1$

$K_2 = Q$

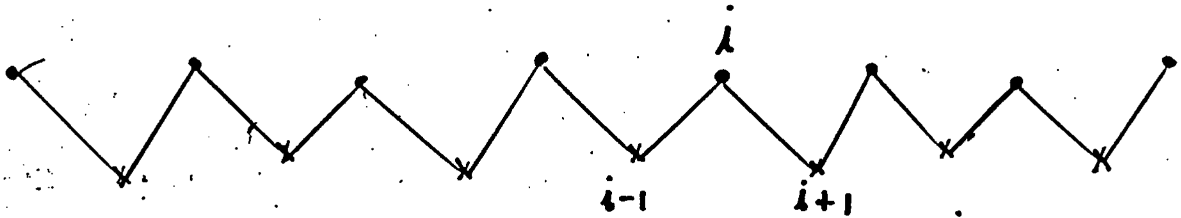
$K_3 = Q^2$

Q

BS	1.0	0.3	0.09	0.027
1	2.2 (-4)	5.4 (-5)	7.3 (-6)	2.3 (-6)
2	1.2 (-3)	8.3 (-4)	3.7 (-4)	1.4 (-4)
3	1.1 (-3)	2.4 (-3)	7.0 (-4)	3.0 (-4)
4	2.6 (-3)	2.6 (-3)	2.2 (-3)	1.4 (-3)
5	4.6 (-3)	4.6 (-3)	3.8 (-3)	3.3 (-3)
6	8.8 (-3)	9.4 (-3)	9.2 (-3)	8.9 (-3)

$\mathcal{E}$

INTERPOLATIONS



3D chain

- fine only

x coarse

(x_i, y_i, z_i) - coordinates of point i

$$\begin{pmatrix}\delta x_i\\ \delta y_i\\ \delta z_i\end{pmatrix} = A_{i-1,i}\begin{pmatrix}\delta x_{i-1}\\ \delta y_{i-1}\\ \delta z_{i-1}\end{pmatrix} + A_{i+1,i}\begin{pmatrix}\delta x_{i+1}\\ \delta y_{i+1}\\ \delta z_{i+1}\end{pmatrix} +$$

$$A_{i-3,i}\begin{pmatrix}\delta x_{i-3}\\ \delta y_{i-3}\\ \delta z_{i-3}\end{pmatrix} + A_{i+3,i}\begin{pmatrix}\delta x_{i+3}\\ \delta y_{i+3}\\ \delta z_{i+3}\end{pmatrix} + \dots$$

A - 3x3 coefficient matrix

LOCAL SETS

For deriving interpolation coefficient

Minimize energy locally:

local relaxations

direct solution

Local set is a truncated chain

Ignore out-of-set links

Facilitate accurate smooth motions

Size of local set

Larger for strong angle (2D) and torsion (3D)

Small remotest interpolation matrix

Effect of local set size in 2D

$K_1 = 1$

	Local set size	
K_2	5=3+2	9=5+4
0.01	0.05	0.03
0.1	0.31	0.09
1.0	0.48	0.12
2.0	0.48	0.1
5.0	0.52	0.11
10.0	0.56	0.20

Asymptotic Convergence Factor of V(1,1) cycles

Energy terms:

$k_1(r - r_0)^2$ - bond interaction

$K_2 r_0^2 (\theta - \theta_0)^2$ - angle interaction